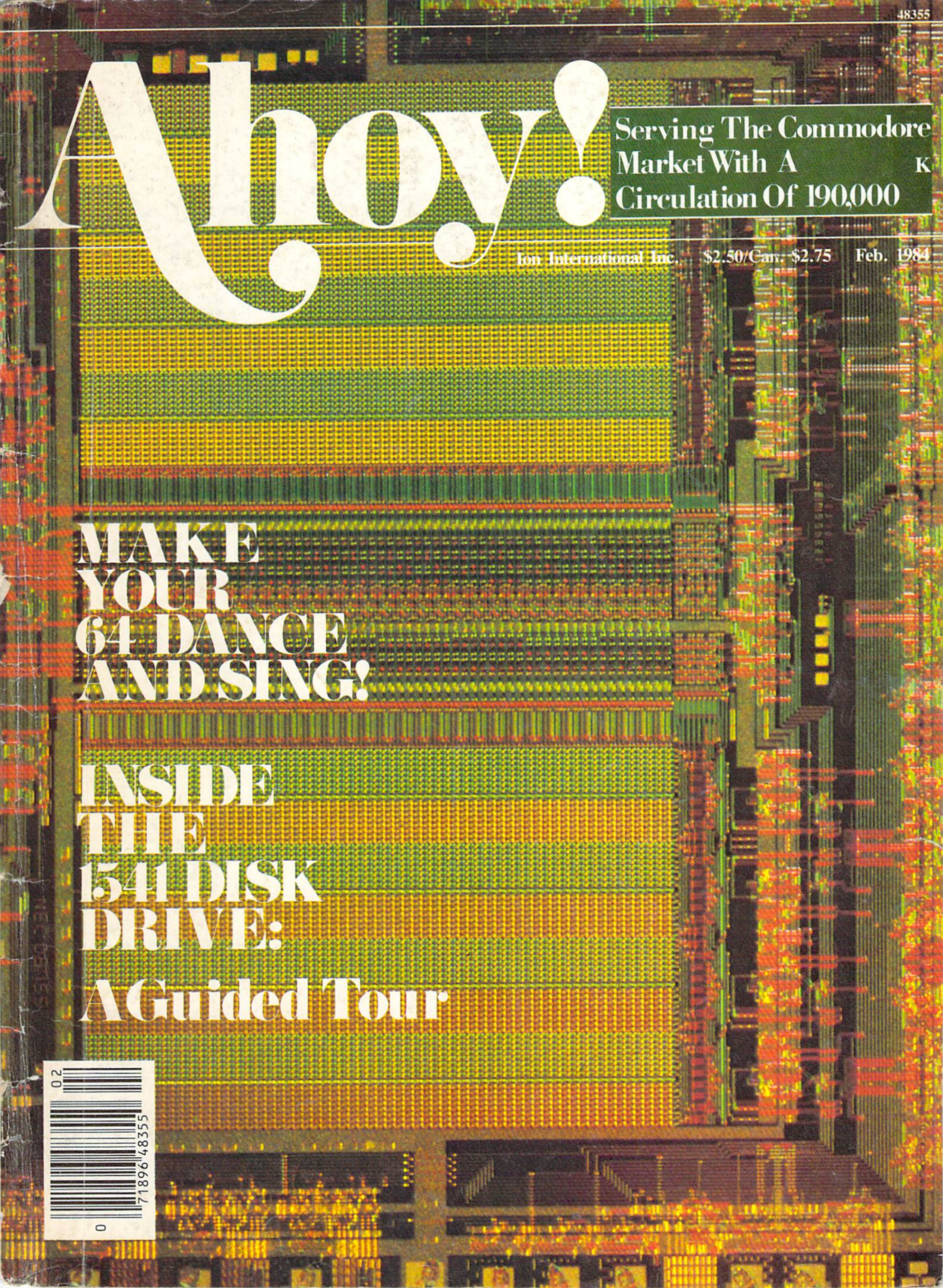
48355

Ahoy!

Serving The Commodore Market With A Circulation Of 190,000

Ion International Inc. $2.50/Can. $2.75 Feb. 1984



MAKE YOUR 64 DANCE AND SING!

INSIDE THE 1541 DISK DRIVE: A Guided Tour

0 71896 48355 02

Ahoy!

CONTENTS

DEPARTMENTS

FEATURES

PROGRAMS

Editor
Steve Springer

Managing Editor
Robert J. Sodaro

Senior Editor
Tim Moriarty

Consulting Editors
Ben Bova
Morton Kevelson
Dale Rupert

Art Director
Joan Mazzeo-Weinshank

Art Production
Eve Griffin
Peter Neiburg
Raoul Tenazas

Publisher
Michael Schneider

Advertising Representative
JE Publishers Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038
(213)467-2266
In Colorado call (303)595-4331

AHOY! No. 2, February 1984. Published monthly by Ion International Inc., 45 W. 34th St., Suite 407, New York, NY, 10001. Subscription rate: 12 issues for $15, 24 issues for $28 (Canada $19 and $36 respectively). Application to mail at 2nd Class postage rates is pending at New York, NY and additional mailing offices.

No responsibility can be accepted for unsolicited material. All editorial and subscription inquiries and software and hardware to be reviewed should be sent to 45 W. 34th St., Suite 407, New York, NY 10001

Cover Photo ©1982 Ross M. Horowitz (the uncredited photographer of our premier issue's cover as well)

PACIFIC COAST SOFTWARE CORPORATION

Commodore 64 software retailing from under $10 to under $100.

The Business Letter Library Series™

A new program for your Commodore 64 system which includes 164 business letters on four disks professionally written for business, administrative and personal use. Ready to customize, print and mail. Compatible with Pacsoft's own Script 64 or PCS 6480 word processors as well as other word processors such as Paper Clip, Word Pro, and Quick Brown Fox. Select the appropriate letter for any situation using the fast LetterSearch feature. These business letters are written to get results! An invaluable tool for any office or home. Categories of letters included in the Business Letter Library are:

Reminders	Complaints
Sales	News Releases
Employees	Surveys
Acknowledgments	Claims
Apologies	Thank yous
Hiring	Requests
Appointments	Reservations
References	Transmittals
Bids	Memos
Meetings	Credit
Orders	Reports
Confirmations	Promotions
Job applications	Agendas
Recommendations	

Also available for the IBM PC, DEC Rainbow and Apple Computers. The Business Letter Library Series is a registered tradename of Delta Point Corporation manufactured exclusively for the C-64 by Pacific Coast Software Corporation.

PCS 6480 80 Column Upgrade includes:

The WORDprocessor, a highly-rated **advanced** word processor with **full** editing and formatting. Automatic margins, centering, right justification, indentation, paging, headers, footers, printer pitch, underlining, boldface, italics, superscripting, subcripting, file linking, list merge, numeric tabs, column setting and manipulation with simple spread sheet functions, five string search and replace, and more! Compatible with most printers. PCS 6480 also includes INQUIRE PAC **and** CALC PAC, infra. Requires Data 20 Video Pak 80.

PCS SCRIPT 64

A highly-rated advanced 40 and 80 column software based word processor featuring horizontal scrolling. Holds 999 screens of text which can be moved within or between screens. Features small and large "control maps" to automatically format individual screens or any series of screens. Global search and replace, superscript, subscript, true insert and delete modes, list or mail merge, automatic hyphenation with wordstream, 20,000 word already formatted dictionary; also includes list merging database. Compatible with most printers.

INQUIRE PAC

A powerful data base in 40 **and** 80 column versions with conditional and threshold searches and sorts on all 15 fields. 200 records per disk, up to 250 characters in last field. Multiple delete and file collecting functions. Disk and file directory. **List merge** with **any** word processor with sequential file structure.

CALC PAC

A sophisticated electronic spreadsheet in 40 and 80 column versions which interfaces with INQUIRE PAC and The WORDprocessor.

BUSINESS PAC

Highly reviewed general ledger, accounts payable and accounts receivable modules available as a package, or **separately**, at fraction of cost of competitive programs. Easy to use, well-documented. General Ledger holds 99 accounts and 700 transactions per period. Generates P & L, Trial Balance and Balance Sheet. Accounts Receivable generates its own invoices and list merges with any sequential file wordprocessor. Accounts Payable has its own checkwriter! Interfacing inventory module available soon. Used with 1525, Gemini 10 or TP1 Printers.

EDITOR PAC

An inexpensive programmer's aid with search and replace, global changes on character strings, variables, etc., display of all variables and values (except arrays). Renumbering, including "goto" and "gosub", etc. Program merging.

FILE PAC

A file manager with most of the features of the larger INQUIRE PAC.

ACCOUNT PAC

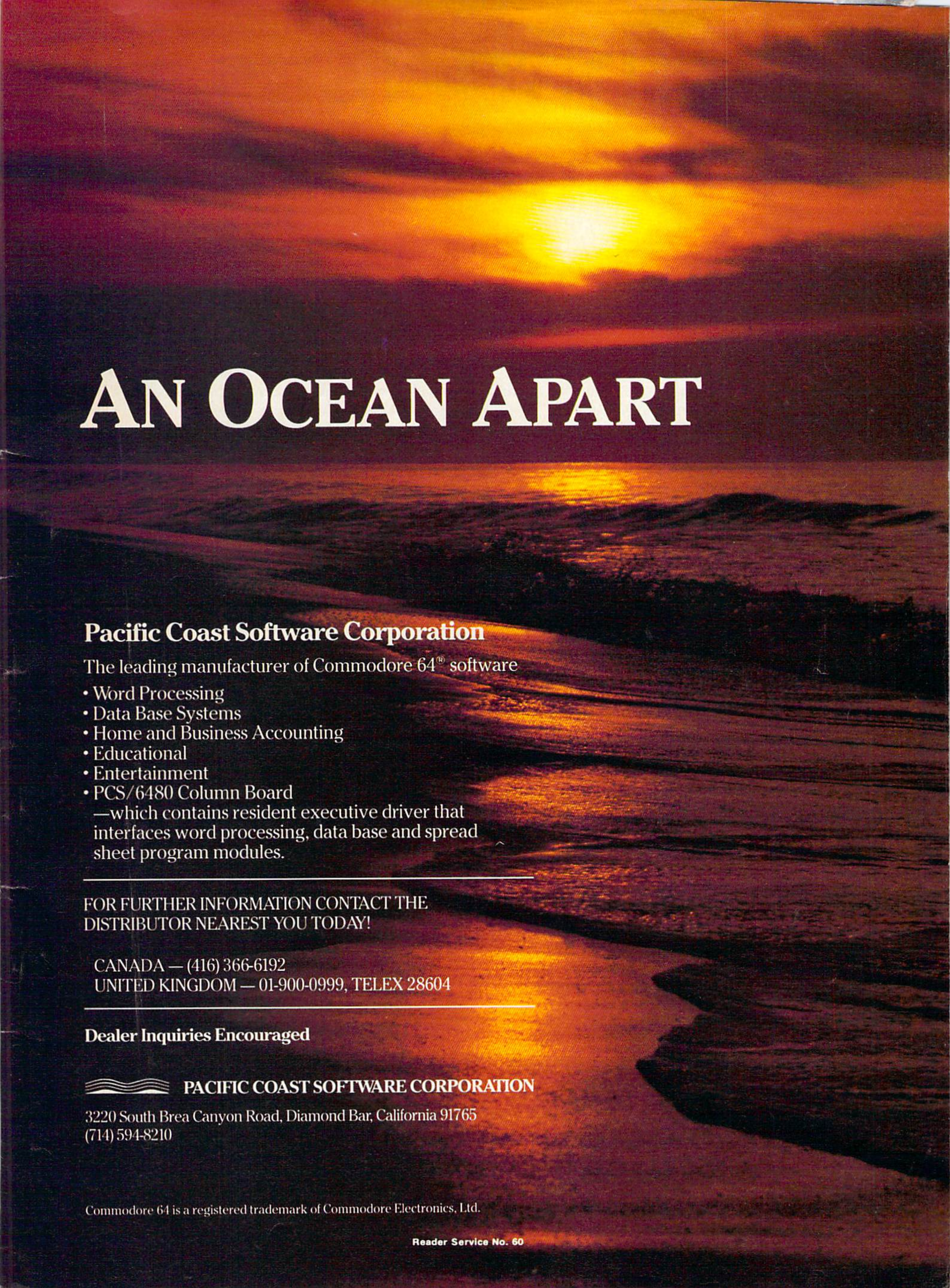
A best-selling, but popularly priced checkbook manager for household or small business. Nearly 100 user-defined acounts with ability to break out expenses and to classify income as taxable or non-taxable. Search and review a single check, an entire account or your entire file, and obtain a hard copy from your 1525 or Gemini 10. Month-end bank reconciliation.

C-64 ASSEMBLER

A highly sophisticated assembler designed for advanced programming. Available soon.

DISTRIBUTORS

Ask about our liberal stock balancing, cooperative advertising, promotional premium plans, upgrades, display stands, promotional literature and customer service representatives (available daily to you and your end-users).


AN OCEAN APART
Pacific Coast Software Corporation
The leading manufacturer of Commodore 64® software
• Word Processing
• Data Base Systems
• Home and Business Accounting
• Educational
• Entertainment
• PCS/6480 Column Board
—which contains resident executive driver that interfaces word processing, data base and spread sheet program modules.
FOR FURTHER INFORMATION CONTACT THE DISTRIBUTOR NEAREST YOU TODAY!
CANADA — (416) 366-6192
UNITED KINGDOM — 01-900-0999, TELEX 28604
Dealer Inquiries Encouraged
PACIFIC COAST SOFTWARE CORPORATION
3220 South Brea Canyon Road, Diamond Bar, California 91765
(714) 594-8210
Commodore 64 is a registered trademark of Commodore Electronics, Ltd.
Reader Service No. 60

EDITORIAL

HOW HIGH IS UP?

It's been said that every new idea evokes three stages of reaction from the general public:

1. "It's completely impossible."
2. "Okay, so it's possible; but what good is it?"
3. "I always thought it would be terrific."

Make that three-and-a-half reactions, because plenty of people say, somewhere between Reaction 1 and Reaction 2, "Even if it is possible, it costs too much."

It's bad enough when the general public reacts that way; after all, most new ideas are based on new information that you and I—the general public—don't have at our fingertips. But all too often the experts, the people who *do* have the new information in their hands, react exactly the same way.

For example: President Truman's military advisor did not believe the atomic bomb would work. The Astronomer Royal of Great Britain refused to accept the idea that rockets could launch satellites into orbit —even after Sputnik! And most computer experts, not so long ago, firmly believed that desktop computers such as the Commodore 64 would not be possible until the end of the century, if then.

Today, the experts and the general public alike have to agree that personal computers are not only possible, they're becoming ubiquitous. Even the book publishing industry, notoriously slow to discover new trends, is now filling bookstores with volumes on how to select and use your very own personal computer.

There are still people who mutter Reaction 1½: they grudgingly admit that personal computers are here to stay, but they complain that they cost too much. Instead of buying and using a personal computer now, they claim they are waiting until the price comes down. While they wait, of course, they are missing all the money-saving advantages of owning and using a computer.

But the biggest grumble over personal computers right now is in the area of Reaction 2: What good is a personal computer? Newspaper articles and television reports talk about the decline of interest in video games, then go on to ask if the boom in home computers has reached its peak.

What good is a personal computer? How high is up?

A Commodore 64, for example, can do everything from word processing to designing airplanes. If you have to balance a checkbook, keep a list of names and phone numbers, write music, study algebra, learn how to spell, or just plain have fun with a videogame, your Commodore 64 is at your service.

But that is merely the beginning. Personal computers are very new, and we haven't yet even scratched the surface of all the things they will be able to do, eventually. Stop thinking of your Commodore 64 as a piece of hardware and start seeing it as it really is —a window on the world.

The screen on which you now write, or do math, or play music, or draw charts is actually an electronic portal that can connect you with a wondrous future. Sooner or later, every library in the world will be linked to your home computer. It will one day monitor your home's heating and air conditioning systems, guard against burglars while you are asleep or away, turn lights on and off in the various rooms of your house, and instantly alert the fire department if and when necessary.

Linked to the telephone, your computer becomes an electronic shopping system, a medical reference, a banking and investment accountant, a calendar that will remind you automatically of important dates and set up the reservations, tickets, and appointments that you require.

In short, the answer to "What good is it?" is simply this: A personal computer is as useful as you make it. It can save you time, money, and effort in a hundred different ways today, a thousand tomorrow, and within a few years, a hundred thousand.

With a personal computer you can bring the world to your doorstep. Your home becomes part of a vast, variegated, endlessly fascinating global electronic village. You—with your computer—become a citizen of the new age.

—Ben Bova

AHOY! *Consulting Editor Ben Bova is one of the world's most respected science fiction writers. The author of more than 63 books, a regular commentator on the CBS Morning News Network television show, and a widely popular lecturer, Bova has also worked as an award-winning editor (*Analog *and* Omni *magazines) and an executive in the aerospace industry. In such novels as* Millennium, Colony, *and* Voyagers, *the Philadelphia-born space activist explores the impact of high technology on individual humans and on society as a whole.*

THE ULTIMATE CONNECTION

INTERPOD FOR COMMODORE 64 OR VIC 20

THE LIMBIC CONNECTION

CONNECTING THE FEATURES OF THE FUTURE FOR TODAY'S COMMODORE USERS.



INTERPOD

Exercise the full potential of the Commodore 64™ or VIC 20™ with the quality, multiple interface from Limbic Systems, Inc. INTERPOD is an intelligent, transparent interface that turns your computer into a powerful system without impacting on your computer's memory.

Simply plug INTERPOD into the serial port of your computer, power-up and you are ready to communicate with any number of IEEE devices and any RS232 printer.

PETSPEED

Speed. . . Ease of Use. . . Ability to compile BASIC programs -- PETSPEED brings the Commodore 4000 & 8000 series computers into the age of action. PETSPEED is the compiler recommended by Commodore. Not just a compiler, PETSPEED contains a powerful optimizer that simplifies the complex, reduces run time and maximizes efficiency of operation.

British Technology for American Results. Interpod and Petspeed are distributed in the United States by Limbic Systems, Inc. Dealer and school discounts are available.

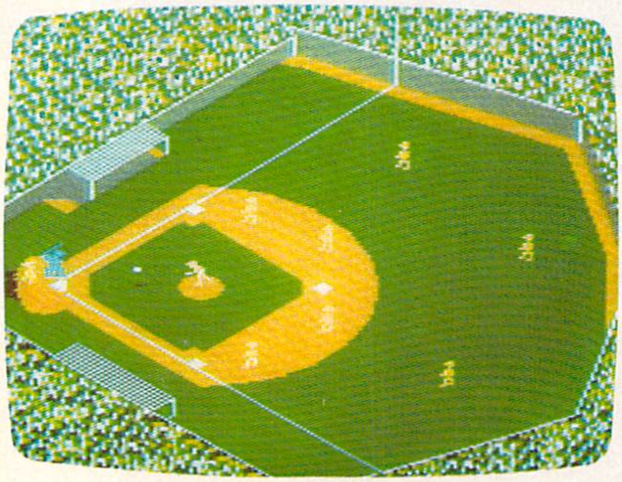
Limbic Systems Inc.

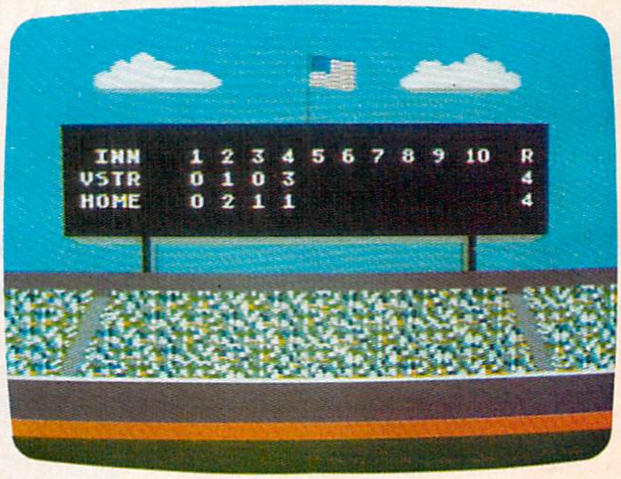
560 San Antonio Road, Suite 202 Palo Alto, CA 94306
(415) 964-8788

Commodore 64 & VIC 20 are trademarks of Commodore Electronics, Ltd.

Star League™ Baseball.

It Makes Dreams Come True On The Atari® 400/800/1200XL™ And Commodore 64.™





If you've ever dreamed of becoming a major leaguer, your dream has just come true. Introducing STAR LEAGUE™ BASEBALL by GAMESTAR. For the Atari® Home Computers and Commodore 64.™

STAR LEAGUE™ BASEBALL brings home all the realistic action *and* strategy of our national pastime. For starters, it offers the most lifelike animation ever seen in a sports game, from the wind-up and delivery of the pitcher to the arc (and shadow) of a fly ball!

STAR LEAGUE™ BASEBALL lets you *choose* your starting team and pitcher. You can even bring in a "knuckleball" throwing reliever when your starter "tires!"

And, of course, STAR LEAGUE™ BASEBALL lets you play *solitaire* against a hard-hitting computer team or a human opponent. All to the exciting sounds of real baseball, from the crack of the bat to the cheer of the crowd!

Ask for STAR LEAGUE™ BASEBALL by GAMESTAR at your nearest Atari® or Commodore dealer today and make *your* major league dream come true. Or write: GAMESTAR INC., 1302 State Street, Santa Barbara, CA 93101 (805) 963-3487 for our *free* catalog and Tips for Stars #2 (for STARBOWL™ FOOTBALL owners).

GAMESTAR™

WE BRING SPORTS ALIVE.

Reader Service No. 39

SCUTTLEBUTT

TUTORIALS • AEROBICS PROGRAM • BUSINESS SOFTWARE • "TED" RUMORS • MATH PROGRAMS • KIDWRITER • 64-APPLE COMPATIBILITY • SMART TERMINAL PROGRAMS • GRAPHICS PROGRAMS • WORD PROCESSORS • TAX PROGRAMS • PCjr • BEACH-HEAD, BUCK ROGERS, AND ALL THE LATEST GAMES

APPLE COMPATIBILITY

Canadian-based Pioneer Software has introduced the Mimic Systems AP Modular Pak, a hardware addition to the C-64 that will allow all Apple II compatible software to run on the 64 and to perform exactly as it would on the Apple.

The Mimic Systems AP Modular Pak has three components:

—The AP bus contains eight standard Apple II peripheral slots and four C-64 expansion slots. It also includes an independent power supply which will drive all the peripheral devices and the 64 itself.

—The heart of the system, the AP "CPU" card, plugs into its own slot on the AP bus and handles all Apple II-to-64 conversions.

—The AP DOS Card turns the 1541 disk drive into an Apple II-compatible drive. The 1541 can then be used for both software systems.

Each of the three elements of the Mimic Systems AP Modular Pak is list-priced at $175. The entire system would then list for $525. Even now we can hear the budgetary pause buttons being pressed: with the 64 becoming a major software target for most publishers, how much quality Apple-only software is out there to justify this purchase?

We envision a time when software crossover will become easy and inexpensive, exploiting the hardware that already exists. Pioneer's Mimic System, if it performs as advertised, is an exciting first step.

Pioneer Software Inc., #217, 620 View Street, Victoria, B.C., Canada V8W 1J6.



Spinnaker's **Kidwriter.**

PEE WEE PROUST

As writers and editors, we naturally are quite enthusiastic about *Kidwriter,* a new C-64 disk for children from Spinnaker. Our own prejudice toward verbal fluency aside, this one looks like a blast.

Designed by Jim and Jack Pejsa and slated for a February release, *Kidwriter* is a storytelling tool which allows children to create their own stories using both pictures and words. The child can choose from a menu of 99 characters to be placed on a variety of colorful backgrounds. Spinnaker lists people, robots, martians, clouds, trees, boats, and cars among some of the available characters. With each new frame, the size, color, and position of the characters can be changed.

Beneath each picture, room is provided to write the story, creating a storybook page. Pages can be saved and called back later in sequence, so that the budding young bard can display his or her book to family and friends.

Spinnaker Software, 215 First Street, Cambridge, MA 02142.

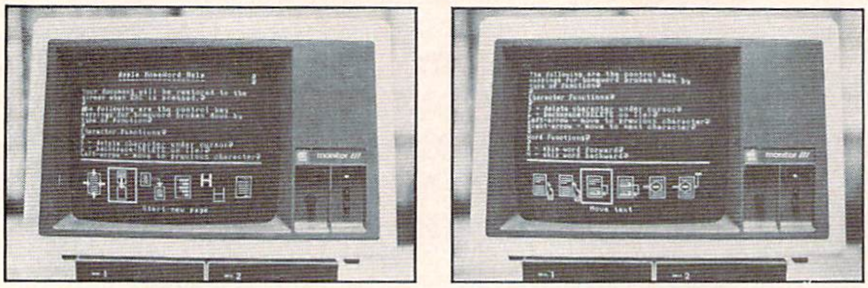
WORD PROCESSORS

After *Kidwriter,* we know that a boring old word processing program with no pictures or martians will sound pretty dry, but perk up: Sierra On-Line's new *Homeword* for the C-64 employs icons, or symbols, of fun stuff like filing cabinets and floppy disks.

Homeword borrows principles first introduced in Apple's Lisa computer. Rather than complex commands or word-heavy menus, the *Homeword* menu displays six icons which represent these options: filing (a filing cabinet is the symbol), editing (a page of print), printing (a printer), format design (an unorganized and organized page with connecting arrows), customizing (a question mark), and disk utilities (a floppy disk).

Within each of these categories, there is a sub-menu offering more specific functions: edit, file, print, layout, customize, and disk utilities.

Homeword divides the screen into three sections. The upper and largest portion of the screen displays the working text; the lower screen portion displays a replica of the entire page as it will print as well as a chart



Homeword's ***symbols make function selection quick and easy.***

which updates the user on available memory and disk space.

Other features include optional joystick control, an outline format, boldface, underlining, file previewing, file merging, headers, footers, and of course the ability to move text portions.

An instruction booklet and audio cassette are included. The package retails for $49.95.

Broderbund has released its *Bank Street Writer* for the C-64 in disk format. The program has long been a best-seller in its Apple and Atari versions.

Every function and command in *BSW* is displayed at the top of the screen for ease of selection. Included among these features: universal search and replace, block move, automatic centering and indent, inverse highlighting of text, word wrap, disk storage and retrieve functions with password protection, redefinable default values, and a print format routine that includes document chaining, page headers and numbering, partial printing, and page break inspection prior to printing.

A tutorial is included on the disk, which will take the novice user through the program's functions. A free backup disk is also included along with documentation. The suggested retail price is $69.95.

Sierra On-Line, Sierra On-Line Building, Coarsegold, CA 93614.

Broderbund Software, 17 Paul Drive, San Rafael, CA 94903.

TUTORIALS

For the don't-know-it-alls among our readership, we present news of a number of recently released programming tutorials.

FlipTrack's *Learning Express* series has developed *How to Operate the Commodore 64.* Two spoken voice cassettes (requiring no computer hookup) and one program data cassette guide the user through steps that will allow him or her to use the sound synthesizer, change onscreen colors and graphics, perform calculations, and program in BASIC. Those with a Commodore Datasette will learn to use the sample programs and load and save their own programs on cassette tapes. The tutorial also includes information on easy use of disk drives and printers. The course is priced at $29.95.

Timeworks has followed up its *Programming Kit I* with *Programming Kits II* and *III.*

Number two is an intermediate level game design and sprite builder. The user is invited to design the game *Slot Machine* while exploring the principles of For/Next loops, arrays, subroutines, special function keys, moving graphics, sound and the use





Programming tutorials by Timeworks, FlipTrack: back to BASIC.

of the RND (randomize) function. A multicolor sprite builder is included with the kit, as are special overlays and documentation.

Programming Kit III is an intermediate level data base system tutorial. A fundamental data base can be designed while aspects of information entry and retrieval, tape storage, sorting techniques, and string arrays are stressed.

Both Timeworks programming kits sell for $24.95.

Timeworks, Inc., 405 Lake Cook Road, Deerfield, IL 60015.

FlipTrack Learning Systems, 999 Main, Suite 200, Glen Ellyn, IL 60137.

EDUCATION

It was certainly naughty of medieval kings to slaughter messengers who brought bad news, wasn't it? Likewise, young children mustn't blame the Commodore computer if it happens to be the means by which they are introduced to math.

Comm*Data, which has been developing Commodore products since 1979, has introduced three C-64 instructional programs in math for children at the preschool or elementary school level.

Toddler Tutor, for preschoolers to second graders, helps children to learn the alphabet, numbers, and colors, and to develop memory skills.

The Primary Math Tutor, for first through fourth graders, provides problems in addition and subtraction.

Math Tutor, for grades three through six, provides instruction in addition, complex subtraction, multiplication and division.

These last two programs offer two levels of difficulty, the second being primarily for drill and practice as it contains no prompting aids or graphic displays.

The programs, each containing a disk and cassette, should be available in most retail outlets.

CBS Software has made its *Success With Math* series available in disk and cassette form for the C-64. The four no-nonsense programs in the series include *Addition/Subtraction* (elementary school students and older), *Multiplication/Division* (for elementary school students), *Linear Equations* (seventh through tenth graders) and *Quadratic Equations* (for students in the eighth through the eleventh grades).

Suggested retail price for each individual disk is $24.95, and for the cassette $19.95.

On an institutional level, Sterling Swift Publishing Company has released the *Fundamentals of Math* series for grade levels three through twelve.

The series was developed by Byron Craig and tested for three years in a Texas school district. The learning system consists of 89 lesson/programs; each lesson includes tutorial and drill & practice, worksheets (which the teacher may copy and foist upon her helpless minions) and documentation.

The programs may be purchased as a six-disk set ($249.95) or as separate sets for third, fifth, and ninth grade levels. Sold separately, they are $69 to $99 each. Worksheets and a preview disk are extra.

Sterling Swift Publishing Co., 791 South IH-35, Austin, TX 78744.

SOFTWARE TO SWEAT BY

While there is a wealth of software that is designed to keep your bank account or stock portfolio in shape, until now there has been nothing to help you keep your body in trim. Spinnaker has provided the remedy: *Aerobics,* designed by the magicians at IPS in collaboration with specialists in dance and exercise. The program is designed for the C-64.



Aerobics' *Fonda-figure.*

With *Aerobics,* the user can select from 18 pre-set exercise segments which vary in length from a half hour to an hour and a half. Exercises are graded for beginner, intermediate or advanced. There are two speed selections, and the workout can be tailored to concentrate on a specific body part or an overall conditioner.

Each segment has four parts: warm-up, aerobics, body parts conditioning and cool down. The graphically represented instructor is rendered in high resolution graphics. Helpful hints and encouragement are presented in captions, and nine tracks of hard-driving music will keep the user motivated.

The suggested retail price of *Aerobics* is $44.95.

Spinnaker Software, 215 First Street, Cambridge, MA 02142.

TWO FROM PIONEER

In addition to the other products mentioned in these pages, busy Pioneer has developed two tutorial packages, one a typing tutorial for the tyro, the other a calculator program for the businessman, each for the 64.

Typing Driller, advertised as

two programs in one, provides drill and practice in typing and word processing skills. For the child, the program can be used as a game.

Pioneer's *Computer Calculator* helps the user take advantage of the math and scientific functions of the C-64. *Computer Calculator* allows the user to design and store complicated mathematical formulas, simplifying difficult or repetitious calculations.

Pioneer Software Inc., #217, 620 View Street, Victoria, B.C., Canada V8W 1J6.

GRAPHICALLY YOURS

Are you more comfortable with a joystick than a paint brush? Are you a businessman with a presentation to give, or perhaps a hacker with an artistic soul that burns to breathe free? If the latter, you've got problems, bub, and three new graphics programs for the 64 won't help.

From Pioneer Software the *3-D Graphics Development Package* is touted as the ideal tool for the architect or draftsman. With a minimum of keystrokes, even a novice programmer should be able to create two- or three-dimensional graphics. And, says Pioneer, with but one additional tap of a key, the perspective can change radically: one minute the user is looking at the house that has been rendered and the next minute the user is inside the house looking out a window.

Adam Bellin, the 19-year-old former Atari programmer, has designed *Sorceror's Apprentice* for Network Search in New York. The program boasts 16 colors, auto fill-in, automatic lines, rectangles, circles, squares and ellipses. Additional features include microview (close up details) and memory move (for duplication).

Pioneer Software, #217, 620 View Street, Victoria, B.C. Canada V8W 1J6.

Network Search, 153 East 32nd St., New York, NY 10016.

INSIDE OUT

In our last issue, David Stone provided the basic information users need to set up terminal programs in their homes. He noted some of the more prominent smart and dumb programs available, such as OMNICOMM and VTE 40. Now our intelligence network has turned up two new and promising terminal programs.

The Smart 64 Terminal from

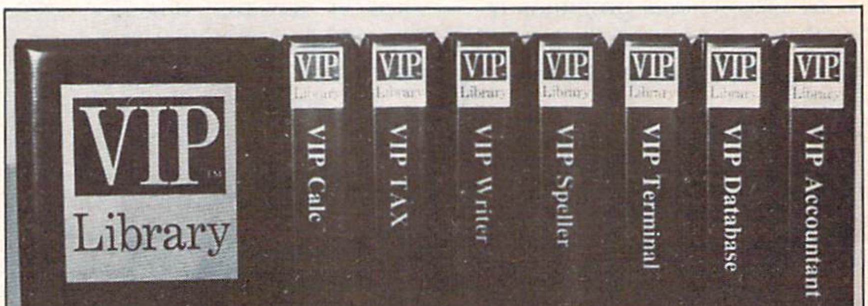


Smart program from SuperTerm.

Microtechnic Solutions, is a disk-based emulator for the 64 featuring both upload and download support functions; full modem control and flexible transmit/receive tables; automatic transmission suspension on full buffer to prevent data loss; an alarm to remind the user when budgeted on-line time has expired; choice of formatted lines or word wrap. The program produces one continuous download file on disk of any length. A download buffer activity is constantly monitored on-line to ensure total user control. The package comes complete with procedures for upload/download program conversions. Retail price is $39.95.

From Midwest Micro—and for both the 64 and the VIC—comes *SuperTerm,* a program that will link the user with CompuServe, an array of hobby bulletin boards in addition to business and university mainframes.

SuperTerm features include a text editor with many customary word processing functions that



Softlaw's **VIP Terminal** *employs symbols for ease of use.*

can manipulate up to 18.4K of information at once (on-or off-line); the ability to display text in 40, 80 or 132 columns using a side-scrolling technique; continuous on-line printing for owners of parallel printers with Midwest Micro's Smart ASCII Plus interface: off-line printing using selected other interfaces or printers; quick saves of large amounts of incoming information and programs on disk; auto-dial and auto-answer using selected automodems; a stand-alone program which converts downloaded program listings into ready-to-run programs without typing; 52 user-defined function keys, 26 display functions and a wide range of settings for baud rates, parity, wordsize, and other parameters.

SuperTerm comes in the form of a hardware module that plugs into the cartridge port, the disk (or cassette) and complete documentation. It retails for $149.95. Smart ASCII Plus sells for $59.95. The VIC version requires a minimum of 16K expansion.

Softlaw Corporation has announced its *VIP Terminal* for the 64, the first in a series of interactive programs for that computer scheduled to become available in the early part of the year. Others promised include *VIP Writer, Speller, Calc, Database, Tax, Accountant,* and something called *Disk-Zap*. All programs will be compatible with the mouse that Commodore is rumoredly developing.

A mouse? The VIP Terminal, like Sierra On-Line's *Homeword,* employs icons, or symbols, to represent the task options available, which include communications (a telephone is the symbol), the phone directory (a little black book), storage (a diskette), and help menus (a question mark).

Softlaw employs a high resolution screen which allows the user a choice of four displays: either 40, 64, 80 or 106 column format, all with 25 lines per screen. Softlaw maintains that its *Terminal* boasts features that are available only to terminal programs for the larger micros, such features including: auto-dial, auto log-on, auto radial, auto upload and download, auto buffer, virtual-memory disk saves, simultaneous printing and downloading, 20 programmable keys, 15 preset phone numbers, and built-in clock and alarm. The full array of communication parameters is included. Both tape and disk formats can be input or output and the program will work with any printer.

The program, packaged in a three ring binder, is priced at $49.95

Midwest Micro Inc. 311 W. 72nd St., Kansas City, MO 64114.

MicroTechnic Solutions Inc. P.O. Box 2940 New Haven CT 06515.

Softlaw Corporation, 9072 Lundale Ave South, Minneapolis, Minnesota 55420.

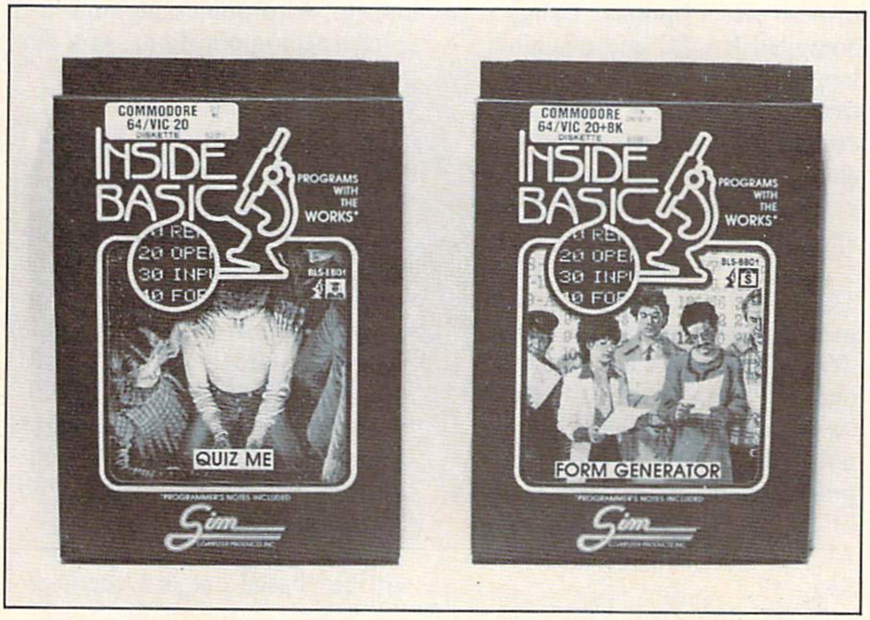
BUSINESS SOFTWARE

Sim Computer Products' first wave of software for the Commodore computers—unreviewed as yet in these pages—at least has the ring of innovation and care.

Each disk or cassette in Sim's *Inside Basic* series includes both the VIC and 64 versions of the program. That series consists of *Quiz Me* (create, load, save and print out your own quizzes) and *Colorcraft* (using basic computer keyboard commands, children sketch and animate their own stories), plus games such as *Kentucky Derby* and *Number Jotto.*

Now Sim has turned its attention to business applications; *Form Generator* and *Home Calc* are the results.

Form Generator allows the user to create his own business forms, invoices, vouchers, statements and labels. The user sets up a master, and then the pro-



Sim Computer Products offers an array of software.

gram will ask for the information needed to complete the form, if need be. A 16K memory expander is suggested for the VIC version. $29.95.

Home Calc, for the 64 only, is described as a user-friendly financial spreadsheet program. Features include sum, replicate, recalculate, title and format, selectable column width and number formats, machine language speed and the custmary mathematical calculation capabilities. Cassette, $24.95.

Meanwhile, TOTL Software has announced the "hatching" of *TOTL Infomaster,* a database management program for the 64, in disk form for $50.

Infomaster is a stand-alone filing system, but it can be used to access files generated by other TOTL programs. The program allows a maximum of ten files per diskette, up to one hundred fields per record, up to 245 characters per field, and up to 2500 characters per record. A repeating field option is provided.

Other features include user-defined report formats and record browse/select options. Using compiled BASIC and machine language routines, disk access time is improved. "Templates" are included in the disk which automatically configure the software for many related data management tasks.

Another recent release from TOTL is *TOTL Speller,* a companion to their word processor, *TOTL Text.* With its proofread option as well as a verify option (which displays suspect words not in its repertoire), the program includes a starter dictionary of 10,000 words, expandable to 24,000. Suggested retail price is $35.

TOTL Software, 155 Third Ave., Walnut Creek, CA 94596.

SIM Computer Products, 1100 E. Hector St., Whitemarsh, PA 19428.

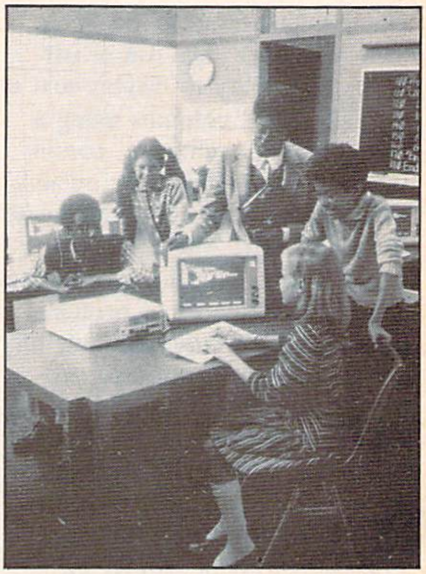
RUMOR, INNUENDO & "TED"

We really hate to print rumor and innuendo in what is essentially a news column. We hesitate to do it, not because we're dedicated journalists and despise rumor-mongering in all its forms (to the contrary on both counts) but rather because, by the time rumor sees print, reality has often confirmed it or dashed it as untruth.

But rumors whirling around the COMDEX show in California and the Commodore show in Toronto late in 1983 are too compelling to ignore. They should be confirmed or dashed at the CES show in early January, at which time Commodore is expected to make its announcement.

That announcement being (rumor has it): that Commodore will release a new computer in 1984. That computer, code-named "Commodore 444" or "The Ted," will incorporate built-in word processing and spreadsheet applications on a single ROM chip. The computer will be comparably priced and equipped to Coleco's Adam (which is selling for $700 to $800).

It will contain a number of innovative new chips, including a video chip. Memory capacity rumors range from 64K to 256K, with most agreeing on a comfortably compromised 128K. It will be one of four new units planned by the company, none of which will be software compatible with the VIC or the 64. Those two computers (rumor! innuendo!) will be phased out in December and March, respectively; meaning, they will no longer be manufactured.



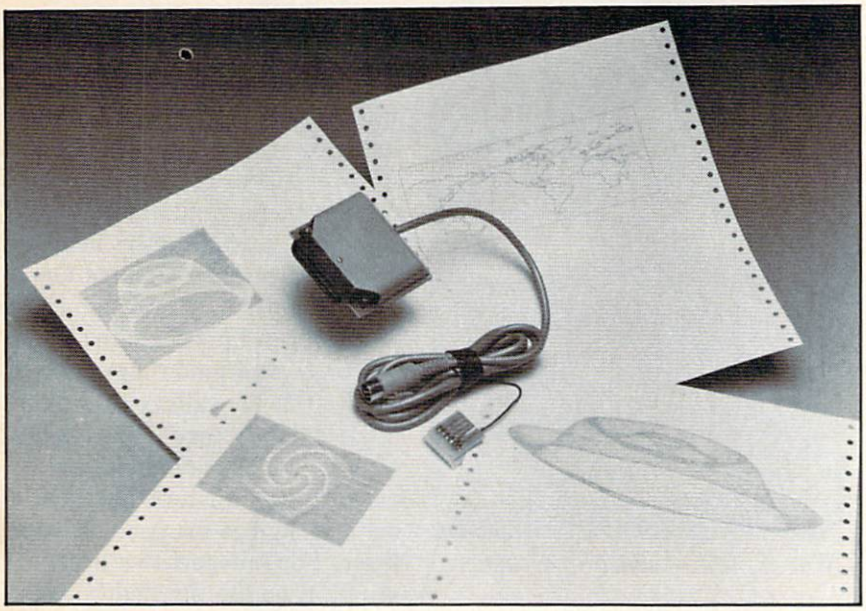
The PCjr: a Big Blue torpedo.

These rumors are based on a report printed in the December 12 issue of *Computer & Software News,* as well as blab gathered by editor Bob Sodaro at the Toronto Commodore show. Commodore president Don Richard denied the accuracy of the reports.

To panic-prone 64 or VIC owners who may feel that supplies of new software will dry up if production ends on those units, we cite the Texas Instruments example. That company's TI99/4A has been discontinued, yet software developers and distributors alike have been coming forward and declaring that they will continue to support the unit. With an installed base in the millions, the Commodore will undoubtedly fare as well or better.

And while we're on the subject of home computers and extinction, this might be a good time to add a postscript to our IBM Peanut article of last issue.

IBM at last unveiled its PCjr, just in time to (IBM hopes) torpedo Christmas sales of rival home computers. Though the unit



The Cardprint/B (C/?B) interface translates nonstandard CBM ASCII.

will not be available until March (realistically, the wait for most individuals will be much longer), IBM announced the product, hoping that most shoppers will delay a computer purchase at Christmas until such time that they can tinker with the PCjr. . . fiddle with its chiclet-style keys.

The PCjr's keyboard is a major disappointment, regressive technology, and is final evidence that IBM indeed did not want to harm sales of the PC with a lower-cost unit. With the rubberized keys, the PCjr is impractical for long-term word processing or programming. Thus, the unit is fine for the businessman who wants to take some light work home, who wants to take advantage of all that IBM-compatible software. But for most home users—gameplayers, education enthusiasts, weekend programmers—the PCjr is not a wise option.

Even as the mainframe, micro and home computer ends of the industry stampede towards IBM-compatibility, IBM-dependence, a backlash is developing. A puny backlash, one which IBM will snicker at. . . but a backlash we applaud. Many companies which support IBM technology and have earned mega-profits from their position are growing nervous that IBM already owns them, for all practical purposes if not on paper. Many of these companies are looking for an exit. Rival business computer companies are taking the risk of ignoring IBM-compatibility. Meanwhile, Christmas consumers are buying non-IBM computers in record numbers.

So three cheers for the independents and those who support them. The sometimes bumbling, sometimes indifferent, ofttimes daring and innovative companies like Commodore and Atari and Apple. And a round of applause for the forgotten man and woman: the consumer who knows how to make a shrewd buy based on price and capability, not on fad or fear.

That's you.

ECONOMIC COMPATIBILITY

Do you want an interface that will translate nonstandard CBM ASCII to any parallel ASCII printer? And to convert the non-printable Commodore graphic characters to a printable two-character code? With the same software compatibility as the Cardprint/A (C/?A), but at a lower price? If you don't, we're sorry we got you this involved. If you do, check out the $49.95 Cardprint/B (C/?B).

When combined with the D/02 Printer Utility package ($24.95), the interface allows printing of hi-res graphics and character graphics, as well as banners in gothic or magtype formats.

TAXING PROGRAMS

It's almost that time of year again. . .and here are a couple of income tax programs that will help you ante up.

Timeworks' *Swiftax* provides 64 users with a menu-driven means of preparing tax returns without prior knowledge of computers or accounting. It guides the user through the tax preparation process, giving instructions and checking tax alternatives automatically, including income averaging, and calculates the lowest amount of tax that must be paid. Other features include printing out amortization schedules and summarizing yearly principal and interest payments. Suggested retail price is $49.95.

Cosmopolitan Software's *Taxpack,* on cassette for the VIC-20, helps Canadian taxpayers do their civic duty for $29.95. A save-and-restore function allows you to record and review historical results, and professional editing features facilitate data entry.

Timeworks, Inc., P.O. Box 321, Deerfield, IL 60015.

Cosmopolitan Software Services Ltd., Box 953 Dartmouth, Nova Scotia B2Y 3Z6, Canada.

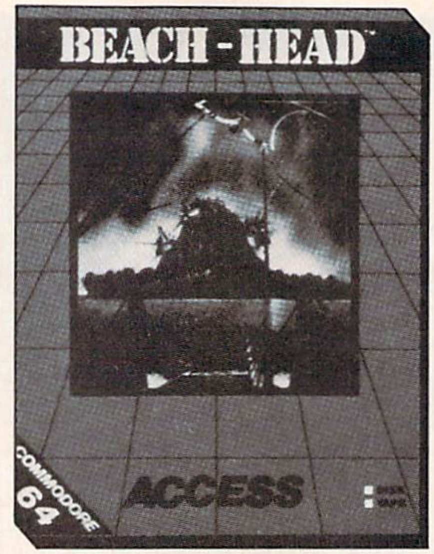
GAMES ROUNDUP

Dickens didn't start producing great works the minute his hands were large enough to hold a quill pen. Like all great writers, he first spent years reading other great writers. So there's no need to feel guilty about shooting up aliens or tracking your way through mazes when your on-line time might be spent honing your programming skills. The concepts you encounter and ideas you acquire playing someone else's program can be invaluable to your development as a C-64 superstar. Of course, Dickens had one advantage over you (at the very least): he was able to glean classical inspiration for free at the public library. You'll have to spring for thirty bucks or so per inspiration. Still, with many libraries now circulating record albums, art prints, films, and videocassettes, perhaps your local booklending institution will soon make it possible for you to study and enjoy the Modern Masters without toiling in a blacking factory.

Addresses of the manufacturers mentioned are provided at the end of the games listings.

Access Software has provided easy access to rock 'em-sock 'em battle action with two new C-64 games.

Beach-Head, on tape or disk from Access, presents the challenge of conquering an island-based fortress that is ruled by a dictator and protected by air, sea, and land forces. As commander of the invasion, you must maneuver your fleet through hidden passages, survive air and sea attacks, and land your amphibious tanks on the beach. You must then thread your way through the island's defense system of land mines, tanks, and anti-tank bunkers up to the fortress. The one-



War is hell—even on the beach.

or two-player game features four 3D screens.

Neutral Zone places you on Alpha IV, a warning station for the detection of alien intruders. With gunnery pod and attack computer at the ready, you prepare to take on a squadron of deadly aliens. All action is in 3D, with five levels of play.

From Comm*Data for the 64: the *Gotcha Math Games,* requiring the player to supply correct answers to arithmetic problems in order to complete such related tasks as blowing up alien spaceships; and *English Invaders,* putting grammatical know-how to much the same purpose.

Datasoft has adapted to 64 format Konami's *Pooyan* arcade game, based on the timeless battle of big bad wolves against little pigs. The player glides up and down on a gondola defending the porkers from hungry wolves who float on balloons, hurling deadly acorns at the player's home. Defenses include shooting arrows at the balloons and throwing chunks of meat to distract the brutes. On another of the multiple screens, the player must stop the wolves from ascending a cliff from which they can roll a boulder down.

As repulsive as they are, even spiders don't deserve to be flattened by rolling apples, drowned in apple juice, mutilated by slicers, crushers, bottlers, and cappers, and scarfed up by hungry frogs and birds. So Sierra On-Line's *Apple Cider Spider* asks you to open up your heart to an arachnid, and guide him back to his web in the rafters of a busy apple cider factory. Who knows? Maybe your altruism will carry over to the next time you spot a daddy longlegs on your kitchen floor. For our part, we'll keep stepping on the disgusting things. For the C-64.

The computer age has provided at least two alternatives to running away to join the circus. You can join the staff of *AHOY!,* and commute to a circus every morning. Or you can play *Sammy Lightfoot,* new for the 64 from Sierra On-Line. Your job is to maneuver Sammy the acrobat across screens of trampolines and trapeze ropes, negotiating rolling barrels, pounding hammers, grinning pumpkins, disappearing floors and tongues of fire. As Sammy completes each stage, he jitterbugs to famous fifties tunes. With perfect timing, he will reach the magic carpet ride that awaits at the end.

The Dark Crystal, utilizing characters, visuals, and plot sequences from the movie, finishes this issue's list of new Sierra On-Line products for the 64. Ah, but for the VIC-20, they've released nine games, seven of which are adaptations and two spanking new.

Ultima II: Escape from Mt. Drash has nothing to do with makeup—on the contrary, it involves escaping from a dungeon through a maze of subterranean

corridors. The split screen format provides the player with simultaneous bird's-eye and floor-level looks at the action in progress. Working against him are an imposed time limit and the monsters he encounters along the way.

Flip-n-Match requires the play er to demonstrate a memory almost on a level with that of his VIC as he looks behind boxes, memorizes the shapes he finds, and tries to locate their twins. In fact, we'd imagine that the game requires a great deal of. . . concentration.

Adaptations include *Cannonball Blitz* (dodge projectiles as you scale a steep hill to a Redcoat castle), *Jawbreaker* (using a set of teeth, chomp your way through a maze of moving walls), *Threshold* (clear trading routes of enemy spaceships), *Crossfire* (blow away invaders that approach from all directions), *Lunar Leepers* (load your crewmen onto your ship before they're eaten by massive-beaked monsters), *Creepy Corridors* (traverse twisting passageways, collecting diamonds and avoiding creatures), and *Frogger* (hippity-hop across a river and a highway without winding up a green splotch on someone's radial).

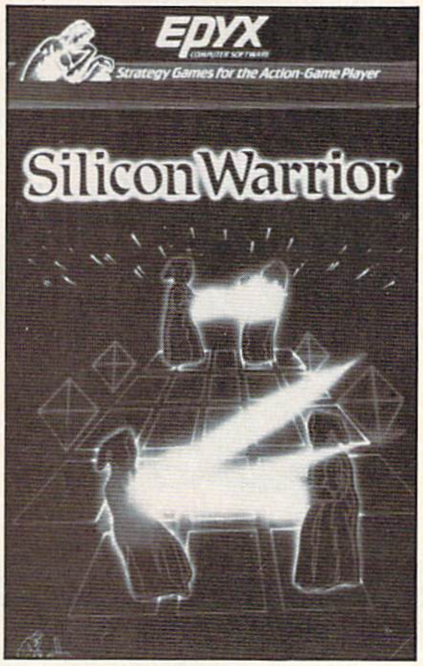
As the great, great grandson of *Zorlok* the wizard, you have inherited a quest: to enter his castle, wipe out a plague of monsters, and regain his treasures. (Why couldn't the old coot just leave Confederate war bonds?) Featuring multiple skill levels, the MicRo Information Systems game is available on tape or disk for the VIC-20.

Three new C-64 games from Spinnaker Software:

Jukebox consists of a 20-square nickolodeon-encased grid. With a pair of dancing feet, you hop from box to box, making records appear and jumping on them to make them grow larger. Eventually, they will turn gold—but only if you jump when the flashing gold squares appear. One misstep, and the music's over.

Alf in the Color Caves requires four to seven year olds to wiggle a wormlike character down through a crosshatch of colorful tunnels. If the Wufflegumps intercept Alf, it's back to the top of the maze. If he makes it, he changes colors and dances to a tune. Spinnaker has released *Alf* as part of its Early Learning Series, citing the child's opportunity to learn "the basic skills of navigation, form and shape recognition, and prediction." Hmm—that extends the definition of "educational" to every videogame ever manufactured.

Lastly, *Trains* puts you behind the pince-nez of a late nineteenth century business tycoon, managing an old-time railroad servicing a network of industries in the Southwest. You must pay bills, set priorities, and meet deadlines as you utilize outside and natural resources to keep things on track.



Valley of the duels.

As Quicksilva's *Quintic Warrior,* the owner of a cassette-driven 64 must stand alone against the sinister Crabmen and a domed city gone mad somewhere in the distant future.

The following eight are the latest Epyx releases for the C-64:

Silicon Warrior chronicles the conflict among the Houses of Apple, Adam, Peanut, and Pong in the year 2084. The goal is a completed computer program that will unravel the mysteries of the universe; the winning warrior must be the first to program five chips on a 3D power grid in a vertical, horizontal or diagonal row. In cartridge format for one to four players.

Gateway to Apshai, the sequel to *Temple of Apshai,* combines role playing and strategy in a quest to collect treasures. While lining your pockets in this one-player game, you'll have to deal with a hellish host of monsters, traps, damsels in distress, and the ubiquitous dungeons. Cartridge format.

Jumpman Junior, sequel to *Jumpman,* requires the player to defend headquarters from infiltration, ducking bullets and overcoming robots, dragons, birdmen, and flying saucers, plus such inconveniences as crumbling girders and vanishing escape routes. The cartridge, playable by one to four persons, features twelve different screens and eight speeds.

Packaged on a single disk or cassette are *Starfire* (attack enemy fighters with lasers while following a message panel that indicates speed, direction, firing ability, and score) and *Fire One* (use periscope and sonar scan to sink enemy fleet while avoiding enemy sub). Both are one-player games.

SYNAPSE EXCITEMENT



On patrol
Out of the sun comes your RAF biplane, loaded down with a deadly cargo of bombs and bullets. But watch out for the anti-aircraft guns and the enemy fighters—a hit could mean a tricky landing for repairs and ammo. **BLUE MAX.***





Ancient treasure
A fortune is yours for the taking. But can you avoid the ghost of Rama and the evil mummy? Are you nimble enough to leap the chasms and outsmart the booby traps between you and freedom? **The PHARAOH'S CURSE.***





Spellbinding
Only you can restore the forest through ancient spells. Then you must march your army of enchanted trees into battle against the Troglodytes and the evil Necromancer. Who will emerge triumphant from the final conflict? **NECROMANCER.***





Take the controls
Your helicopter mission—capture vital fuel and weapons, free the enslaved masses, and finally destroy the fortress itself. Will you triumph or be crushed by the fiendish Kraalthan lords? **FORT APOCALYPSE.***

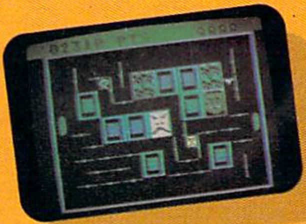


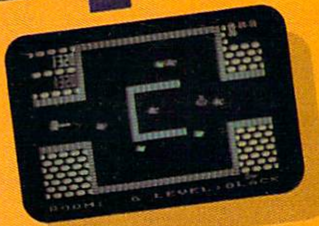
*Trademarks of Synapse Software.
Commodore 64 is a registered trademark of Commodore Inc.

FOR YOUR C-64!





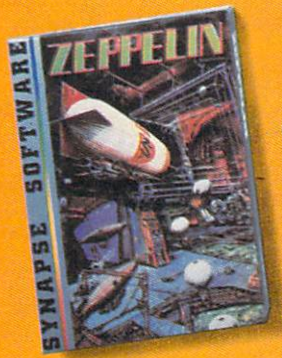




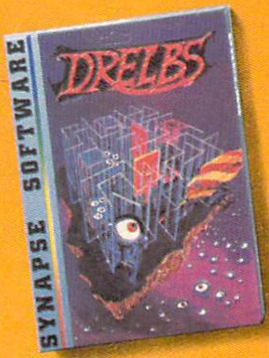
Awesome action
Maybe you've played pinball before, but not like this! No time to think, no room to make even one mistake. Just quick reflexes, light body armor and a whole lot of luck between you and the end of the game.
SLAM BALL.*



Very hot air
First the prison break, but that's only the beginning! The underground world of Zarkafir is full of surprises, from the lethal energy fields to devastating earthquakes. Can you defeat the Timelords?
ZEPPELIN.*



Flip-flop
Into this miniature land comes the evil Trollaboars, determined to take over. Their screwhead tanks will surely crush the peaceful Drelbs, unless you can defeat them on the atomic flip grid. **DRELBS.***



The Shadow knows
Deep in his lair the Shadow waits, protected by deadly Robo-Droids, Whirling Drones and Snap-Jumpers. Only the very strong and the very quick are ever seen again! **SHAMUS* & SHAMUS CASE II.***



Write for a FREE catalog and Elite Club information.

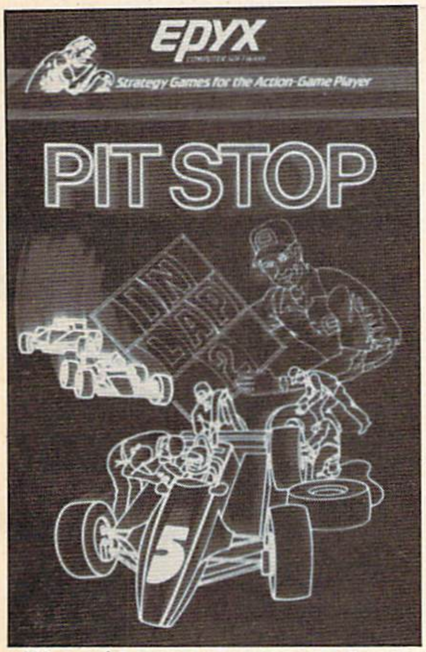
Synapse games are also available on disk and cassette for the Atari, Apple and IBM home computers.

Name ____
Address ____
City ____
State ____ Zip ____
Computer ____ C

synapse SOFTWARE

5221 Central Avenue, Richmond, CA 94804 · 415/527-7751

Reader Service No. 62



Wrenching motor race action.

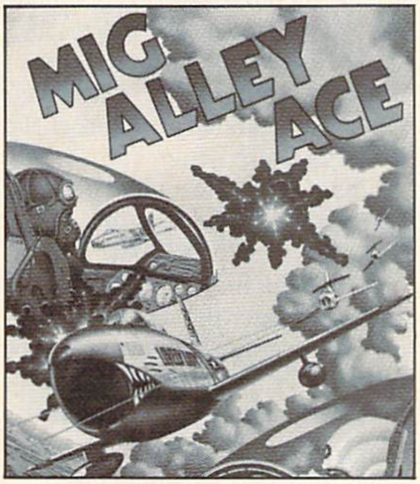
Pitstop thrusts race-game fans into the real world, where it's not all whipping around corners and weaving through the pack. You will have to enter the pit periodically to refuel and retool. When you enter is your decision, based on factors such as how much gas you use and how fast your tires wear out. Can be played by one to four people.



900 multiple choice questions.

QUESTION: Which of the following statement(s) about *Fax* is true? A) It is a coin-op game from Exidy; B) It has been released by Epyx for the 64; C) The computer version features nine hundred multiple choice questions in sports, trivia, history, and entertainment, with three levels of difficulty; D) The game can be played by one person or by two players who race to supply the correct answer first; E) It is available on disk or cassette. ANSWER: All of the above.

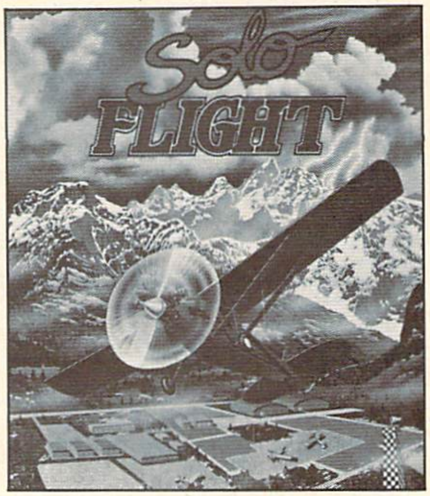
If you're in a martial mood, MicroProse Software has adapted to C-64 format four games that would quench even George S. Patton's bloodthirst. *MIG Alley Ace* is a full-scrolling real time



Full-scrolling real-time wargame.

wargame that requires players to react quickly to combat reports during a Soviet Bloc invasion of Europe. You can use air power, tactical nuclear weapons, and combat forces from all the NATO countries to battle the Red Threat. *Solo Flight Simulator* lets you practice takeoffs, landings, cross-country navigation, and emergency procedures in Day, Night, Crosswind, and Instrument Flying scenarios. *Hellcat Ace* is a three-dimensional aerial dogfight above the Pacific. All of the above come on disk or cassette.

Not exclusively a children's program, *Fun With Art* lets the user draw freestyle or choose



Puts you in the pilot's seat.

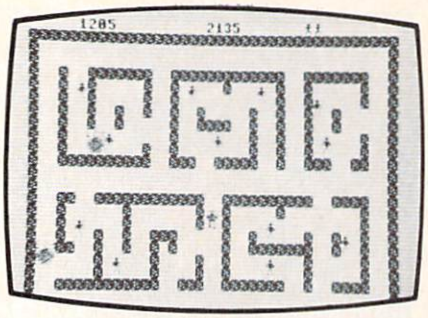
from a menu of brush strokes, shapes, and 128 colors which he may combine. On cartridge.

Fun With Music requires the player to compose a song, then try to play the tune in game format without missing a note. On cartridge.



A dogfight in a hellcat--beastly!

Brand new from Muse Software is *Rescue Squad,* requiring you to pluck unfortunates from a blazing building. And newly adapted is *Castle Wolfenstein,* embroiling a captured allied soldier in espionage and adventure in an ancient fortress. Both for the C-64.



Rescue Squad: ***be a hero.***

In *Wildcatting,* one of five from Image Computer Products for the C-64, you will try to find a hidden oil deposit, taking into account a geological survey and the per meter cost of drilling. The computer creates a different deposit each game.

If you strike it rich, *Wall Street Challenge* will let you invest in a variety of stocks, ranging from steady earners to risky high-flyers. Stock charts and the Dow Jones will assist you. 8K and 16K versions are included.

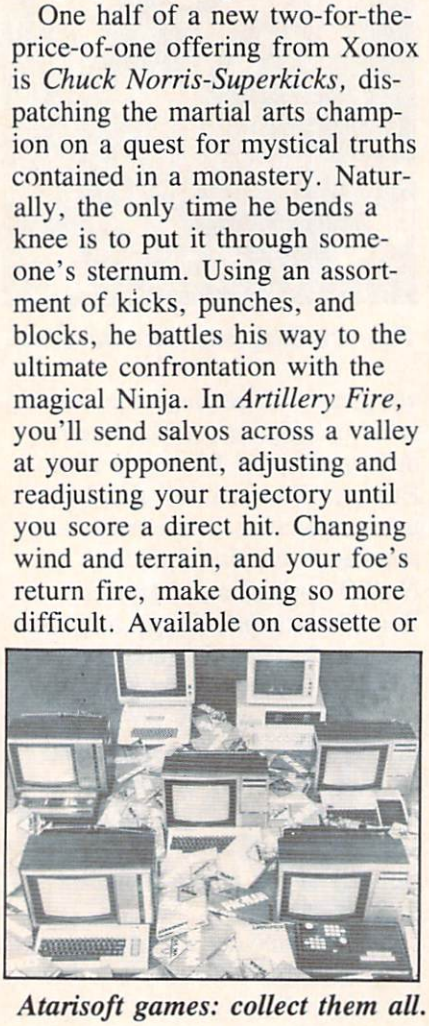
Completing the Image: *Bouncer* (bounds from one trampoline to another, clearing off squares and avoiding arrows that burst him), *Romeo* (traverse a scorching desert, a stream swimming with alligators, sharks, and floating logs, and treacherous terrain), and *Diablo* (a 116-panel contest that requires you to keep an ever-advancing ball from rolling off the board).

Broderbund's *Sky Blazer* charges VIC owners with five different missions to clear away enemy radar, tanks, and ICBM installations, while evading heat-seeking missiles, explosive balloons, and enemy jets. Also adapted to the VIC by Broderbund: *Lode Runner,* putting you in the boots of a Galactic Commando who must run, jump, drill passageways, and outfox life-threatening guards across 24 different game screens.

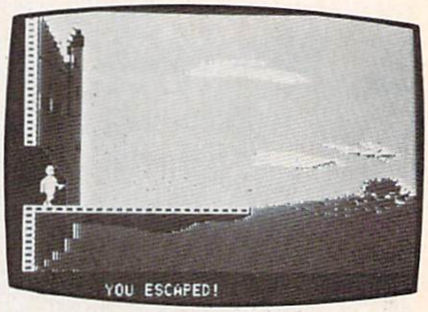
Reston has adapted *Miner 2049* onto cartridge for both the VIC-20 and C-64. Players must joystick-guide Bounty Bob on his quest for lucre, avoiding radioactive mutants who roam the shafts, fatal falls, and miscalculated jumps, while trying to claim all the mining stations (seven screens in the VIC version, ten in the 64) in the allotted time.

Going from riches to rags, *Infidel* from Infocom puts the 64 user in the sand-filled shoes of an explorer marooned by his followers in the middle of the desert. Your only hope in this totally textual adventure is to reach the great lost pyramid that you came to Egypt to find—and that's when the skullduggery and cliff-hanging really begin.

One half of a new two-for-the-price-of-one offering from Xonox is *Chuck Norris-Superkicks,* dispatching the martial arts champion on a quest for mystical truths contained in a monastery. Naturally, the only time he bends a knee is to put it through someone's sternum. Using an assortment of kicks, punches, and blocks, he battles his way to the ultimate confrontation with the magical Ninja. In *Artillery Fire,* you'll send salvos across a valley at your opponent, adjusting and readjusting your trajectory until you score a direct hit. Changing wind and terrain, and your foe's return fire, make doing so more difficult. Available on cassette or disk for the C-64.



Atarisoft games: collect them all.



Castle Wolfenstein ***for the C-64.***

Synapse has adapted *Zaxxon,* claiming that their version will take full advantage of the 64's graphics capabilities.

Figuring that one giant alien menace would not satiate 64 owners, Commodore Software has adapted Bally Midway's *Lazarian.* The game's three missions include rescuing a ship from a swarm of meteors, traveling down the multilevel Tunnel of Fear, and facing off with the menacing galactic leviathan for whom the game is named.

Atari has released its first batch of adaptations for the VIC and 64, consisting of *Dig Dug* (burrow horizontal and vertical tunnels in search of fruits and vegetables—and avoid the monsters they conceal), *Defender* (save helpless humanoids from an array of creatures on an alien world), *Stargate* (*Defender's* sequel), *Robotron 2084* (rescue humans from robot monsters bent on mass murder), *Centipede* (blast) the garden pest snaking its way toward you), *Donkey Kong* (roll out the barrels), and *Pac-Man* (scarf up the dots).

Sunrise has slated for release four action games for the 64. In *Quest for Quintana Roo,* Yucatan Sam seeks to solve a Mayan mystery by crawling through the hundreds of chambers that compose the mystical temple of Quintana Roo. In *Rolloverture* you

help a conductor scurry around the orchestra pit, placing music balls in music maker slots to insure that the correct notes are played. The educational *Campaign '84* requires you to plot your Presidential campaign, including defining your stand on the issues and raising funds. The goal of *Gust Buster* is to land in an amusement park where you will sell balloons, avoiding rides, fireworks, elephants, and other hazards.

Two arcade adaptations from Sega for both the VIC and 64: *Buck Rogers—Planet of Zoom* requires you to defend the 25th century world from an alien onslaught. In day and night scenarios, you must skim the surface of the planet, navigating through electron posts while battling swooping alien saucers, space hoppers, and, ultimately, the alien Mothership in the trackless void of outer space.

As the hunter in *Congo Bongo,* you must climb Jungle Mountain and cross a chasm and a river to reach and corral the mighty ape. Points are earned for the number of hazards crossed and the number of screens completed within a specified time.

Adapted for the C-64 by Sirius are *Wavy Navy* (dodge bombers and Kamikaze fighters on the high seas) and *Critical Mass* (jet around the world piecing together clues to save the earth's five largest cities from exploding).

Access Software Inc.
925 East 900 South
Salt Lake City, UT 84105
(801) 532-1134

Sunrise Software Inc.
2829 W. Northwest Hwy.
Suite 904
Dallas, TX 75220
(214) 352-3999

Datasoft, Inc.
9421 Winnetka Avenue
Chatsworth, CA 91311
In CA: (213) 701-5161
Rest of USA: (800) 423-5916

Sierra On-Line, Inc.
Sierra On-Line Building
Coarsegold, CA 93614
(209) 683-6858

Broderbund Software, Inc.
17 Paul Drive
San Rafael, CA 94903
(415) 479-1170

Spinnaker Software
215 First Street
Cambridge, MA 02142
(617) 868-4700

Image Computer Products, Inc.
P.O. Box 3761
Cherry Hill, NJ 08034
(609) 667-2526

Reston Publishing Co., Inc.
11480 Sunset Hills Rd.
Reston, VA 22090
(703) 437-8900

Infocom, Inc.
55 Wheeler Street
Cambridge, MA 02138
(617) 492-1031

Xonox—USA Sales Office:
K-tel International Inc.
11311 K-tel Drive
Minnetonka, MN 55343
(612) 932-4000
Canadian Sales Office:
K-tel International Ltd.
K-tel International Ltd.
1670 Inkster Blvd.
Winnipeg, Manitoba R2X 2W8

Sirius Software, Inc.
10364 Rockingham Drive
Sacramento, CA 95827
(916) 366-1195
In CA: (916) 366-1195
Rest of USA: 1(800) 824-4222

Quicksilva, Inc.
426 West Nakoma
San Antonio, TX 78216
(512) 340-3684

Epyx, Inc.
1043 Kiel Court
Sunnyvale, CA 94089
(408) 745-0700

MicroProse Software
10616 Beaver Dam Road
Hunt Valley, MD 21030
(301) 667-1151

Muse
347 N. Charles Street
Baltimore, MD 21201
(301) 659-7212

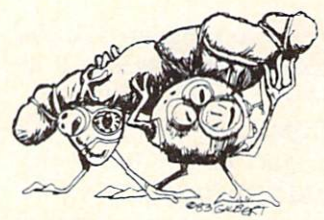
Synapse Software
5221 Central Avenue
Richmond, CA 94804
(415) 527-7751

Commodore Software
1200 Wilson Drive
West Chester, PA 19380

Atari Incorporated
1265 Borregas Avenue
P.O. Box 427
Sunnyvale, CA 94086
(408) 743-4810

Sega Consumer Products
5555 Melrose Avenue
Los Angeles, CA 90038
(213) 468-4800

Comm*Data Computer House
320 Summit
Milford, MI 48042
(313) 685-0115



By Pete Lobl

The 64 Graphics System

PART II

This edition, I'm going to do two things. First, I'll provide a loader program which will allow you to load Multi-Draw without having to use all those POKE's. Second, I'll present the disassemblies of the graphic routines so you can examine them and hopefully learn what they do. They are well-documented, but next time I will discuss them further. Here goes:

```
10 POKE53280,0:POKE53281,0:PRINTC
HR$(14)CHR$(8)
20 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{
CD}{WH}PLEASE WAIT, MULTI-DRAW 64
 IS LOADING..."
30 PRINT"{HM}{CD}{CD}{CD}{BK}POKE
44,64:POKE64*256,0:NEW
40 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}
{CD}LOAD"CHR$(34)"MULTI-DRAW 64"C
HR$(34)",8
50 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{CD}{CD}RUN{HM}
60 POKE631,13:POKE632,13:POKE633,
13:POKE198,3
```

The above program will load in Multi-Draw without your having to do all the POKE's and load by yourself.

Type in the loader program and SAVE it. Whenever you want to use Multi-Draw, simply say LOAD "LOADER",8 or ,1 depending on whether you use disk or tape. Run this program and it will boot up Multi-Draw and run it for you. It eliminates a lot of hassle.

On the subject of the disassemblies, I'm now going to show you the routines that Multi-Draw uses in its operation. If you examine PLOT 1, PLOT 2, PLOT 3, and PLOT 0, you will notice that they use almost the identical code—there are very few differences among them. We've listed all four in their entirety, to make it easy for you to cross-refer and spot the differences. All the other routines—FILL, READ JOY, etc.—are pretty straightforward and you can incorporate them into your own programs with little difficulty.

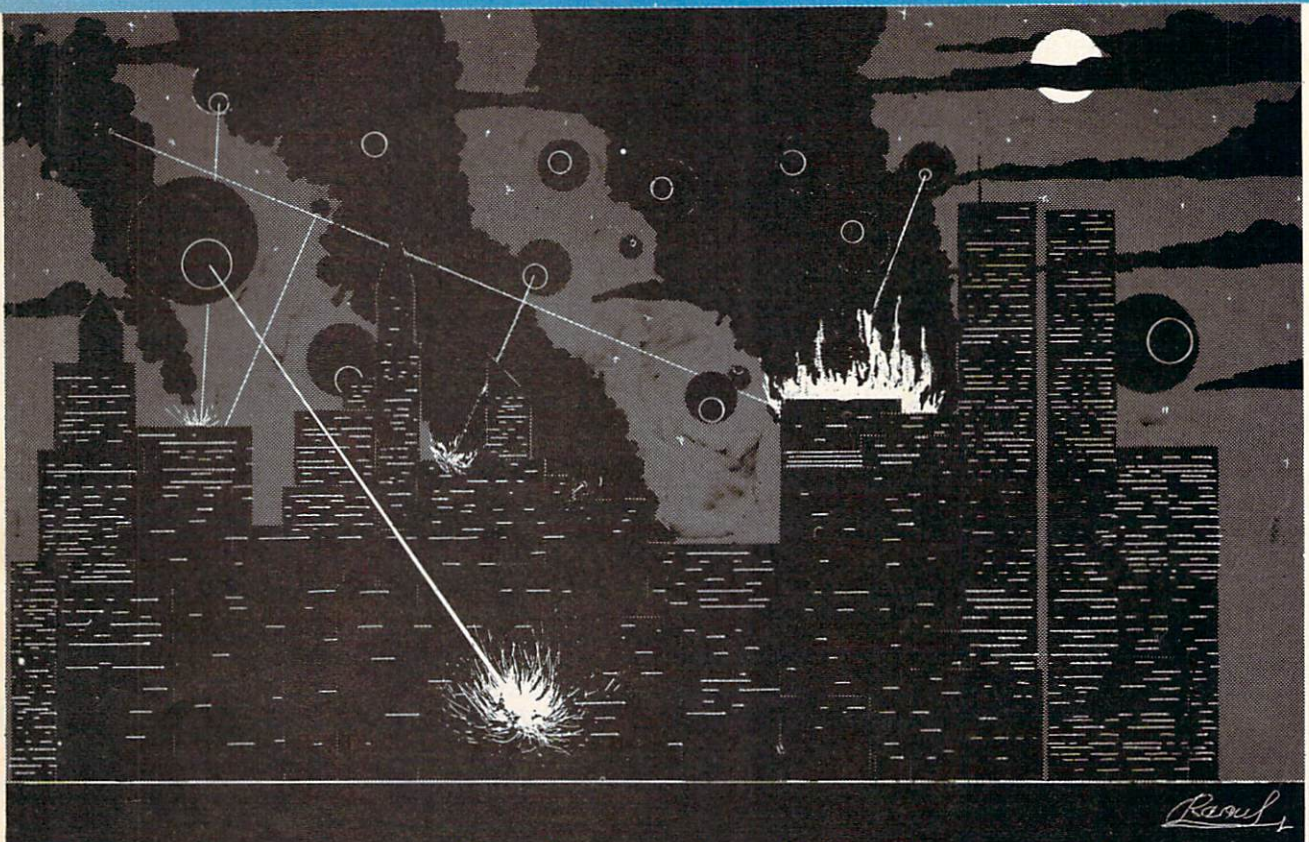
(Last month, I said we would incorporate 10 new commands into the Multi-Draw system. I think I jumped the gun a little, and will wait until the current Multi-Draw system is understood totally. You can't run before you walk, right?)

Running short on allotted space since the disassemblies take up quite a bit of room, I will close this edition of the Graphics System. In the next installment, we will take an indepth look at what Multi-Draw does and how its machine code operates. Also, we will look at the Raster-Scan Interrupt, a power of your 64 with enormous potential. Between the split screening and the use of 64 sprites, the overall value of the RSI will become evident. Till we meet again, happy hacking!

(IMPORTANT! The Multi Draw listings in the back are for reference only. *Do not type them in as a BASIC program.* Only with an assembler or a comparable monitor can these be entered. In any case, these will show you how machine language looks and how it is written.)

SEE PROGRAM LISTING ON PAGE 82

By Robert Alonso



NIGHT ATTACK

(For the VIC-20)

Early in the 21st century, a spaceship armada lifted off from the planet Nova. Its destination was earth; its mission, to subjugate the planet, rich in the natural resources Nova required.

Our world's defenses, primitive by comparison, were by no means adequate to turn back the invasion—not without deploying our most powerful nuclear weapons, certain to kill millions of humans along with the invaders. But thanks to the foresight of the scientists of a previous generation, earth was saved from enslavement.

Years earlier, a holograph generator had been erected in the middle of an uninhabited stretch of African wasteland. When the Novan attack was sighted, the image that had been programmed into the generator's computer memory was projected: that of a gleaming, spired city hundreds of miles wide, with towers stretching almost into the clouds. The invasion commander, naturally presuming this to be earth's capital city, ordered his entire fleet to converge on the site.

As the warships fired futilely at the phantom city, nuclear warheads converged on them from all directions. They exploded above the uninhabited territory, injuring no one—no one except the Novan fleet, destroyed to the last ship.

To commemorate the victory, a glistening city was built on the site of the battle—a smaller-scale representation of the ghost metropolis that had saved the planet. It was christened Nova City.

It took the Novans years longer to rebuild their invasion fleet. With that done, they have returned—their primary goal being to level the city that serves as a reminder to the entire galaxy of their humiliating defeat at the hands of a mirage.

This time, it's a fight to the end—and only you can save the people of Nova City. Five waves of vengeance-crazed aliens will attack the city; if you're dexterous enough, you will defeat them, and

encounter the ultimate wave of Nova attackers. If you survive this wave, your city will be completely repaired. However, the drones' attack will be relentless, and you will have to survive more consecutive waves.

To play the game, you will have to begin by typing in the first section of the program and saving it on tape. Type in the second section and save it right after the first. This is done to conserve memory on the VIC-20. The first program loads in the character set and then loads in the game program from cassette. Once you have saved both programs, rewind the tape and load the first section. Run it, and read the instructions. The second program will be automatically loaded. As soon as the game is ready to run, you will be prompted to hit K for keyboard control. If you are using a joystick, just hit any other key; but if you're using the keyboard, just hit the K key and you're on your way to doing life-and-death battle with ambulatory aliens! All the instructions, including which keys to use, are detailed in the instruction screen of the first program.

Important: you must to use the ''Abbreviations for BASIC Keywords'' listed in your user's manual.

Some details you should be aware of: you get ten points for every hit and twenty-five points for every direct hit. Once you have accumulated five hundred points a siren will go off and you will meet another wave of aliens (they're shaped differently). For every five hundred points you accumulate after that, you will hear the same siren go off and yet another wave appear. All aliens that you miss will damage your city and once the damage is substantial the city will blow up in flames and the game will end.

Good luck and have lots of fun! If you prefer not to type in the program and would like to have it ready to run, send a blank cassette and a self-addressed mailer with five dollars to:

Nova Soft (AHOY!)
Box 527
Nutley, NJ 07112

A verified copy will be sent within two days.

SEE PROGRAM LISTING ON PAGE 80

PROGRAMMING RELATIVE FILES ON THE COMMODORE 64

By Michael Kleinert and David Barron

Last month we talked about the sequential disk file. This month we continue our series on disk files by discussing the *relative* file type.

As we stated last month, data in sequential files is accessed one item after another; hence the name "sequential." A relative file can be thought of as one large file containing many smaller sequential files.

In a sequential file, in order to get to a place in the file all data previous to the item you desire must be read in. Contrary to this, in a relative file, you may jump directly to a particular place in the file. This place is called a record.

Each record is a miniature sequential file that may range up to 254 characters. When a relative file is created, the user must specify how long each record is, or its record length. The computer will then set aside this amount of space for each record. There may be up to 720 records in a single file. All these numbers add up to the fact that the only limit on the size of a relative file is the amount of data that may be stored on a disk.

Here is a diagram of a typical relative file. In this file, each record is 10 characters long and contains a name. There are a total of 7 records.

```
SAMPLE RELATIVE FILE
          B B B B B B B B B B
          Y Y Y Y Y Y Y Y Y Y
          T T T T T T T T T T
          E E E E E E E E E E
          # # # # # # # # # #
          1 2 3 4 5 6 7 8 9 10
                  ---------------------
RECORD 1 :  /H/A/R/R/Y/+/ / / / /
                  ---------------------
RECORD 2 :  /J/A/M/E/S/+/ / / / /
                  ---------------------
RECORD 3 :  /J/O/E/+/ / / / / / /
                  ---------------------
RECORD 4 :  /D/A/V/I/D/+/ / / / /
                  ---------------------
RECORD 5 :  /T/I/M/+/ / / / / / /
                  ---------------------
RECORD 6 :  /M/I/C/H/A/E/L/+/ / /
                  ---------------------
RECORD 7 :  /T/O/O/D/L/E/+/ / / /
                  ---------------------
```

You may have noticed that each of the names ends in a plus sign (+). This represents a data terminator. It is usually a carriage return. These are present so that only the pertinent data is read in, and the trailing characters in each record are ignored.

Suppose that we were curious as to what was stored in record number 7. We would simply instruct the computer to look at record 7 and it would tell us that "TOODLE" was stored there. If this were a sequential file, we would have to read in all of the preceding data to get to item number 7. As you are probably beginning to realize, relative files can make life a whole lot easier, and they allow greater versatility in data manipulation.

At this time, we'll bet you're wondering how to apply this in your programs. Well, we'll not keep you in suspense any longer.

Relative files must be created a bit differently from sequential file types. An OPEN statement is

used to initially create a relative file. The format of this OPEN statement is as follows:

```
OPEN(FILE#),(DEVICE#),(CHANNEL#),
"(FILENAME),L,"+CHR$(X)
```

Let's dissect the above statement:

OPEN—Tells the computer to create a file with the following information.
(file#)—Can be any file number that is not currently open, ranging from 1 to 127.
(device#)—Specifies what I/O device is being accessed. In this case the device number is the 8, the disk drive.
(channel#)—Can range from 2 to 14 for data files. For other purposes, numbers 0, 1, and 15 have been reserved.
(filename)—Can be any name up to 16 characters long. This is what the file will be called in the directory.
'',L,''—Specifies that a relative file is being created. The '',L,'' is omitted when accessing a previously created relative file.
CHR$(X)—The value of X specifies the record length, ranging up to 254. This is also only specified when initially creating a relative file.

The following lines of BASIC are examples of creating relative files.

```
10 OPEN 3,8,3,"EXAMPLE,L,"+CHR$(5
0)
```

```
10 OPEN A,B,C,A$+",L,"CHR$(D)
```

The first line above will create a relative file named ''EXAMPLE'' with a record length of 50. The second line will create a relative file with a name that is the contents of A$ and a file length of the value of D.

Once a relative file is created, a slightly easier format is used to OPEN it again. You need only specify its name; the '',L,'' and +CHR$(X) are omitted. For example, to access an existing relative file called DEMO, you might use:

```
OPEN 3,8,3,"DEMO"
```

The disk operating system will see that this is a relative file, and it will take care of the rest of the details. (*Note:* the replace option will not scratch and recreate a relative file.)

Once a relative file has been created, you may write data into it or read from it. Unlike sequential files, you do not have to specify whether you will be reading or writing when it is opened.

Before reading from or writing to a relative file, you must open the command channel with OPEN 15,8,15 and then tell the DOS (Disk Operating System) which record you wish to access data from. The format for this is as follows:

```
PRINT#15,"P"CHR$(CHANNEL#)CHR$(LO
)CHR$(HI)CHR$(POS)
```

(channel#) should be identical to the one used in your OPEN statement. ''lo'' and ''hi'' tell the DOS which record number you wish to access. Use the following formula to determine the values for lo and hi where R is equal to the record number.

```
HI=INT(R/256):LO=R-(HI*256)
```

(pos) determines how many bytes into the particular record data will start being read from. To start from the first character in the record, simply omit CHR$(pos) or make (pos) equal to 1.

For example, let's say we wanted to look at data starting at the first character in record number 6; we would use the following statement:

```
PRINT#15,"P"CHR$(C)CHR$(6)CHR$(0)
```

where F is the file number and C is the channel number. To access record number 260 starting at the 5th character, we would use:

```
PRINT#15,"P"CHR$(C)CHR$(4)CHR$(1)
CHR$(5)"
```

Once you have specified the record number, you may begin writing data into it with the PRINT# command. An example would be PRINT#3,A$. Similarly, you may read from a record with the INPUT# and GET# commands. You may not read or write past the end of a record into the next record.

Possible disk errors resulting from I/O with relative files are as follows:

50 RECORD NOT PRESENT—This error may occur when specifying a record number past the end of the file. If you are simply expanding the file, you may ignore this ''error'' message. This error message will also occur if you read past the end of the last record with the GET# or INPUT# statements.

51 OVERFLOW IN RECORD—This error condition means that you have attempted to write past the end of a record with a PRINT# instruction. As a result, the information being written to the record is truncated so that it fits within limits.

52 FILE TOO LARGE—This error condition indicates that you have specified a record number that is too large and the disk cannot hold that many records of the specified size.

The phone book program that follows demonstrates the usage of relative files as we have dis-

Continued on page 78

TELSTAR 64

Sophisticated Terminal Communications Cartridge for the 64.

PFO 10D 00D CP D1 D2 BELL 12:30:00 10:14:36
(TELSTAR's Status Line)

Don't settle for less than the best!

- Upload/Download to/from disk or tape.
- Automatic File Translation.
- Communicates in Industry Standard ASCII.
- Real-Time Clock plus Alarm Clock.
- Line editing capability allows correcting and resending long command lines.
- 9 Quick Read functions.
- Menu-driven.
- Similar to our famous STCP Terminal package.
- Works with Commodore Modems and supports auto-dialing.

The best feature is the price — **only $49.95** (Cartridge and Manual)

Machine Language Monitor Cartridge
for the CBM 64

More than 20 commands allow you to access the CBM 64's Microprocessors Registers and Memory Contents. Commands include assemble, disassemble, registers, memory, transfer, compare, plus many more.

Someday every CBM 64 owner will need a monitor such as this.

Cartridge and Manual — **$24.95**

Reader Service No. 52

8K in 30 Seconds
for your VIC 20 or CBM 64



If you own a VIC 20 or a CBM 64 and have been concerned about the high cost of a disk to store your programs on ... worry yourself no longer. Now there's the RABBIT. The RABBIT comes in a cartridge, and at a much, much lower price than the average disk. And speed ... this is one fast RABBIT. With the RABBIT you can load and store on your CBM datasette an 8K program in almost 30 seconds, compared to the current 3 minutes of a VIC 20 or CBM 64, almost as fast as the 1541 disk drive.

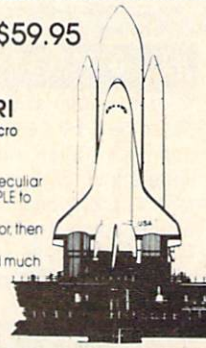
The RABBIT is easy to install, allows one to Append Basic Programs, works with or without Expansion Memory, and provides two data file modes. The RABBIT is not only fast but reliable.

(The Rabbit for the VIC 20 contains an expansion connector so you can simultaneously use your memory board, etc.)

$39.95

MAE NOW THE BEST FOR LESS!

$59.95



For CBM 64, PET, APPLE, and ATARI

Now, you can have the same professionally designed Macro Assembler/Editor as used on Space Shuttle projects.

- Designed to improve Programmer Productivity.
- Similar syntax and commands - No need to relearn peculiar syntaxes and commands when you go from PET to APPLE to ATARI.
- Coresident Assembler/Editor - No need to load the Editor, then the Assembler, then the Editor, etc.
- Also includes Word Processor, Relocating Loader, and much more.
- Powerful Editor, Macros, Conditional and Interactive Assembly, and Auto - zero page addressing.

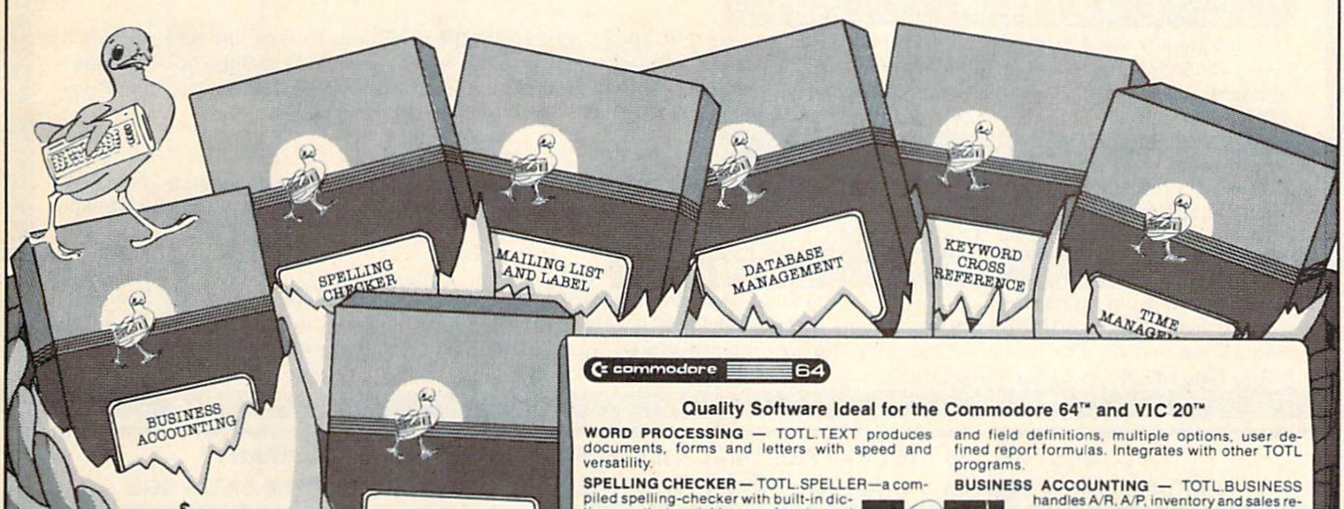
Still not convinced; send for our free spec sheet!

Eastern House

3239 Linda Dr.
Winston-Salem, N.C. 27106
(919) 924-2889 (919) 748-8446
Send for free catalog!

VISA® MasterCard

A Basket Full of Software Values at Chick-Size Prices? Eggs-Actly!



commodore 64

Quality Software Ideal for the Commodore 64™ and VIC 20™

WORD PROCESSING — TOTL.TEXT produces documents, forms and letters with speed and versatility.

SPELLING CHECKER — TOTL.SPELLER—a compiled spelling-checker with built-in dictionary that quickly proofreads and suggests correct spelling.

MAILING LIST AND LABEL — TOTL.LABEL organizes mailing lists, collection catalogs, menus, all listing and sorting. Merges with TOTL.TEXT.

DATABASE MANAGEMENT — TOTL.INFOMASTER features dynamic record and field definitions, multiple options, user defined report formulas. Integrates with other TOTL programs.

BUSINESS ACCOUNTING — TOTL.BUSINESS handles A/R, A/P, inventory and sales reports and other record-keeping tasks.

TIME MANAGEMENT — TOTL TIME MANAGER plans schedules and analyzes events in numerous ways.

KEYWORD CROSS REFERENCE — RESEARCH ASSISTANT—a great tool for automated indexing and cross reference system.

TOTL SOFTWARE, INC.
quality you can afford
1555 Third Avenue
Walnut Creek, CA 94596
(415) 943-7877

Reader Service No. 53

COMMODORE 64
(more power than Apple II at half the price)

$99.50*

- 170K DISK DRIVE $159.00*
- TRACTION FRICTION PRINTER $109.00*

COMPUTER AND SOFTWARE SALE

WE HAVE THE BEST SERVICE | WE HAVE THE LOWEST PRICES

VIC-20
(a real computer at the price of a toy)

$69.50

- 40-80 COLUMN BOARD $59.00
- 32K RAM EXPANDER $95.00

* COMMODORE 64 COMPUTER $99.50
You pay only $199.50 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your computer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net computer cost is $99.50!!

SOFTWARE BONUS PACK $29.95
When you buy the Commodore 64 Computer from Protecto Enterprizes you qualify to purchase ONE SOFTWARE BONUS PACK for a special price of $29.95!! Normal price is $49.95 (40 programs on disk or 24 programs on 5 tapes).

* 170K DISK DRIVE $159.00
You pay only $259.00 when you order the 170K Disk Drive! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your disk drive that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net disk drive cost is $159.00.

* TRACTION FRICTION PRINTER $109.00
You pay only $209.00 when you order the Comstar T/F deluxe line printer that prints 8 1/2 x 11 full size, single sheet, roll or fan fold paper, labels etc. 40, 66, 80, 132 columns. Impact dot matrix, bi-directional, 80 CPS. LESS the value of the SPECIAL SOFTWARE COUPON we pack with your printer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied your net printer cost is only $109.00.

80 COLUMN BOARD $99.00
Now you program 80 COLUMNS on the screen at one time! Converts your Commodore 64 to 80 COLUMNS when you plug in the 80 COLUMN EXPANSION BOARD!! List $199 **SALE $99** PLUS—you also can get an 80 COLUMN BOARD WORD PROCESSOR with mail merge, terminal emulator, ELECTRONIC SPREAD SHEET. List $59.00 **SALE $24.95** if purchased with 80 COLUMN BOARD!! (Tape or Disk)

80 COLUMNS IN COLOR
EXECUTIVE WORD PROCESSOR $69.00
This EXECUTIVE WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE for PROFESSIONAL Word-processing application! DISPLAYS 40 OR 80 COLUMNS IN COLOR or Black and White! Simple to operate, powerful text editing with a 250 WORD DICTIONARY, complete cursor and insert/delete key controls line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers! Includes a powerful mail merge. List $99.00 **SALE $69.00.** 20,000 WORD DICTIONARY - List $24.95 **SALE $19.95.** EXECUTIVE DATA BASE - List $89.00 **SALE $59.00.** (Disk only).

SPECIAL SOFTWARE COUPON

We pack a SPECIAL SOFTWARE COUPON with every COMMODORE 64 COMPUTER-DISK DRIVE-PRINTER-MONITOR we sell! This coupon allows you to SAVE OVER $100 OFF SALE PRICES! $200-$300 savings are possible!! (example)

PROFESSIONAL SOFTWARE COMMODORE 64

Name	List	Sale	Coupon
Executive Word Processor	$99.00	$69.00	$59.00
Executive Data Base	$89.00	$59.00	$46.00
20,000 Word Dictionary	$24.95	$19.95	$14.95
Electronic Spreadsheet	$89.00	$59.00	$46.00
Accounting Pack	$69.00	$49.00	$32.00
Total 5.2 Word Processor			
Tape	$69.00	$56.00	$37.00
Disk	$79.95	$63.00	$42.00
Total Text 2.6 Word Processor			
Tape	$44.95	$39.00	$26.00
Disk	$49.00	$42.00	$29.00
Total Label 2.6			
Tape	$24.95	$18.00	$12.00
Disk	$29.95	$23.00	$15.00
Programmers Helper (Disk)	$59.00	$39.00	$29.95
Basic Tutor (Tape/Disk)	$29.95	$24.95	$15.00
Typing Teacher (Tape/Disk)	$29.95	$24.95	$15.00
Sprite Designer (Disk)	$16.95	$14.95	$10.00
Medicinemen (Tape)	$19.95	$17.95	$12.00
Weather War II (Tape)	$19.95	$17.95	$12.00
Professional Joy Stick	$24.95	$15.95	$11.00
Light Pen	$39.95	$19.95	$16.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60

(See other items in our catalog!)

Write or call for Sample SPECIAL SOFTWARE COUPON!

EXECUTIVE QUALITY PROFESSIONAL BUSINESS SOFTWARE

The Cadillac of business programs for Commodore 64 Computers

Item	List	*SALE
Inventory Management	$99.00	$59.00
Accounts Receivable	$99.00	$59.00
Accounts Payable	$99.00	$59.00
Payroll	$99.00	$59.00
General Ledger	$99.00	$59.00

(*COUPON PRICE $49.00)

VIC-20 COMPUTER $69.50
This 25K VIC-20 computer includes a full size 66 key typewriter keyboard color and graphics keys, upper/lower case, full screen editor, 16K level II microsoft basic, sound and music, real time floating point decimal, self teaching book, connects to any T.V. or monitor!

40-80 COLUMN BOARD $59.00
Now you can get 40 OR 80 COLUMNS on your T.V. or monitor at one time! No more running out of line space for programming and making columns! Just plug in this Expansion Board and you immediately convert your VIC-20 computer to 40 OR 80 COLUMNS!! List $129. **SALE $59.00.** You can also get an 80 COLUMN BOARD WORD PROCESSOR with mail merge, terminal emulator, ELECTRONIC SPREAD SHEET!! List $59.00. **SALE $24.95** if purchased with 80 COLUMN BOARD! (Tape or Disk).

32K RAM EXPANDER $95.00
This cartridge increases programming power over **8 times!!** Expands total memory to 57K (57,000 bytes). Block switches are on outside of cover! Has expansion port!! Lists for $199 **(OUR BEST BUY!)**

60K MEMORY EXPANDER $49.00
Sixslot — Switch selectable — Reset button — Ribbon cable — CARDCO. A must to get the most out of your VIC-20 Computer!

8K RAM CARTRIDGE $39.00
Increases programming power 2 1/2 times. Expands total memory to 33K (33,000 bytes). Memory block switches are on outside of cover! Includes FREE $16.95 game.

16K RAM CARTRIDGE $55.00
Increases programming power 4 times. Expands total memory to 41K (41,000 bytes). Memory block switches are an outside cover! CARDCO Includes FREE $29.95 adventure game!!

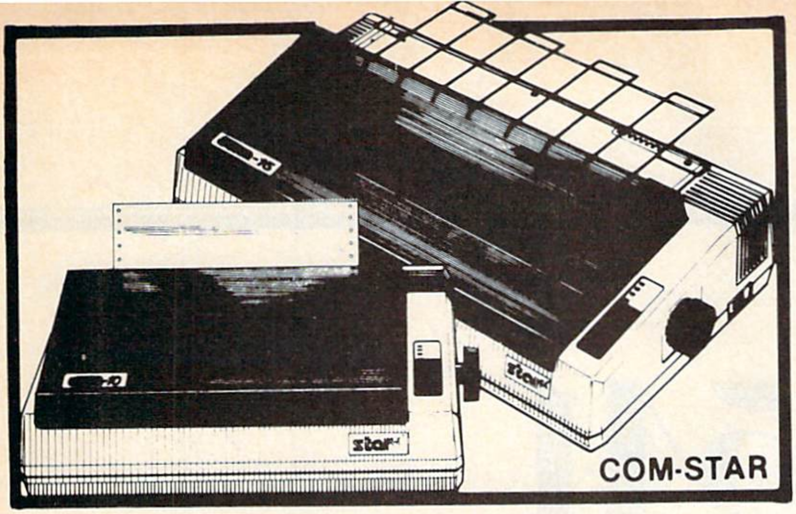
12" GREEN SCREEN MONITOR $99.00
Excellent quality GREEN PHOSPHOROUS VIDEO MONITOR with antiglare, 1920 characters (80 characters x 24 rows). Save your TV! a must for 80 column word processors. PLUS $9.95 for VIC 20 or Commodore 64 Cable.

12" AMBER SCREEN MONITOR $119.00
Premium quality AMBER VIDEO MONITOR With antiglare, (80 characters x 24 rows), exceptionally clear screen, faster scanning. PLUS $9.95 for VIC 20 or Commodore 64 Cable.

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD — C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order



FANTASTIC PRINTER SALE

as low as **$149.00**

• *15 Day Free Trial - 180 Day Immediate Replacement Warranty*

	LIST	SALE
80 COLUMN THERMAL PRINTER — 60 CPS Bi-directional, dot matrix, prints 8½″ letter size paper, full 80 columns, high resolution graphics, dot bit addressable, special symbols and true decenders! (Centronics parallel interface)	$199	**$149**
80 COLUMN TRACTOR-FRICTION PRINTER — 80 CPS Bi-directional, dot matrix, impact, prints single sheets, continuous feed paper, adjustable columns, 40 to 132 characters! Roll paper adapter $32.95. (Serial or Centronics parallel interface)	$399	**$209**
PREMIUM QUALITY 10″ CARRIAGE T/F PRINTER — 120 CPS Bi-directional, impact, 9 x 9 dot matrix with double strike for 18 x 18 dot matrix. High resolution bit image (120 x 144 dot matrix) underlining back spacing, left and right margin settings, true lower decenders, with super and sub scripts. Prints standard, italic, block graphics, special characters, plus 24 of user definable characters and much more!! Prints single sheets, continuous feed and roll paper! (Centronics parallel interface)	$499	**$289**
PREMIUM QUALITY 15½″ CARRIAGE PRINTER — 120 CPS Has all the features of the Premium Quality 10″ Carriage T/F Printer above plus a 15½″ carriage and more powerful electronic components to handle large business forms! (Centronics parallel interface)	$599	**$379**
HIGH SPEED PREMIUM QUALITY T/F 10″ PRINTER — 160 CPS Save printing time with these plus features: 160 CPS speed, 100% duty cycle, 8K buffer diverse character fonts special symbols and true decenders, vertical and horizontal tabs. This is Red Hot Efficiency!!! (Serial or Centronics parallel interface)	$699	**$499**
HIGH SPEED PREMIUM QUALITY T/F 15½″ PRINTER — 160 CPS Has all the features of the 10″ Carriage high speed printer plus a 15½″ carriage and more powerful electronics to handle larger business forms! (Serial or Centronics parallel interface)	$799	**$599**

PARALLEL PRINTER INTERFACES: (IN STOCK)

- For VIC-20 and COMMODORE 64 — **$49.00**
- For all APPLE COMPUTERS — **$69.00**
- For ATARI 400 and 800 COMPUTERS — **$79.00**

NOTE: *Other printer interfaces are available at computer stores!*

SALE SALE

WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

Reader Service No. 45

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

INSIDE THE

1541 DISK DRIVE:

A Guided Tour

By Morton A. Kevelson

The Commodore 64 has been selling like hotcakes and shows no signs of slacking off. Along with the computer, copious quantities of the Commodore 1541 disk drive have been sold as well. In fact, the number of users who have bought drives along with the computer actually took Commodore by surprise and at one point created shortage.

This should really have been expected in view of the well under $300 selling price, which makes the 1541 a real bargain compared to other manufacturers' drives. Furthermore, dealers' shelves have been overflowing with high-quality, sophisticated software for the 64, making tape storage impractical at best.

In view of the proliferation of 1541 disk drives, it is rather surprising that very little has been said about the inner workings of the machine. Commodore releases little or no information and the manual supplied with the drive has achieved a certain notoriety for poor documentation. Speaking from experience, the manual presently supplied with the 1541 is actually a considerable improvement over the version that accompanied its 1540 predecessor.

In the interest of filling this information vacuum, at least to some extent, we will take you on a guided tour of the 1541 disk drive. At the very least, your curiosity about what is inside that mysterious box should be somewhat satisfied. The more adventurous of you will also be able to change the disk device number for multiple drive use and perform basic cleaning chores.

Before we begin, a word of caution. Do not go tearing into your drive at the first sign of trouble or when you first read this article. Most of the activities we will describe here will, at the very least, void your warranty. There are some delicate adjustments in the drive mechanism which should not be disturbed without proper training and tools. Proper head alignment probably requires a special alignment disk and test equipment which most users do not have. Unless you are reasonably adept at things mechanical, we suggest leaving this sort of tinkering to a qualified serviceman.

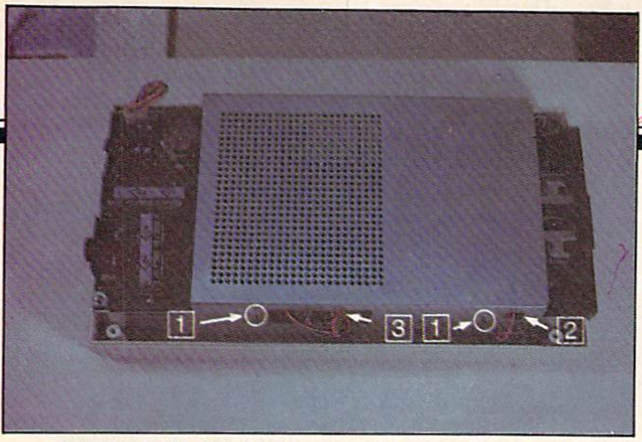


PHOTO 1: Disk drive with cover removed.

To begin with, we have to establish a common reference point. As you might expect, the front of the drive is the end into which you insert your disk, the back is opposite. Left and right will always be referenced with the front of the drive facing you.

You will need a number one Philips head screwdriver. That is the kind that has a crossed tip rather than a single flat blade. Make sure the blade is sharp and in good condition. A worn Philips head screwdriver invariably wears out the screw head, causing no end of grief. A long-nosed pliers or a long pair of tweezers will come in handy for picking up the screws which will fall to the bottom of the case. If you feel you are super careful, you can dispense with the latter. And come up and introduce yourself sometime. I'd love to meet someone who does not drop an occasional screw! A word of caution here: do not use magnetized tools of any sort near the drive.

For cleaning the head, a small bottle of iso-propyl alcohol 91 percent should be obtained at your local drug store. A four-ounce bottle should cost about a dollar. Do not use rubbing alcohol, as this contains additives which will leave a deposit on the head surface. Some good-quality cotton swabs and a can of compressed air or a squeeze blower finish off the list. Do not go blowing into your drive, as human breath contains moisture and other corrosives which the inside of a disk drive is better off without.

We will start by turning off all power to the drive and computer. Disconnect all cables from the back of the drive and turn it over, placing it on a firm, smooth surface. A sheet of cardboard, not corrugated, makes a good protective layer to prevent scratches. At each corner is a rather deeply recessed Philips head screw. You may have to apply considerable pressure to these screws the first time you take them out. The screws are too short to come out of the case, so just leave them in the holes. Hold the drive case together and turn it back over. Some of the screws may fall out at this point. Lift off the top half of the case, poke out the remainder of the screws, and put them aside.

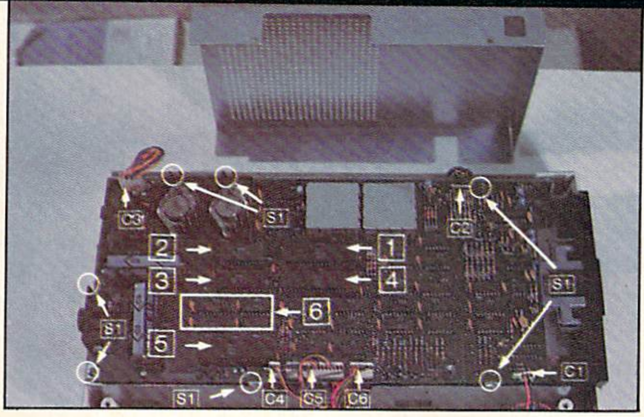


PHOTO 2: Long board (found in older 1541 units).

At this point the metal housing will be visible (Photo 1). This is held in place by two screws on the left side of the housing (item 1, Photo 1). The right side of the housing is held in place by a pair of dimples in the sheet metal. Before removing the housing, carefully examine the wires from the green power on indicator light emitting diode in the front right corner. If the wires are routed through the small cutout towards the front (item 2, Photo 1) you will have to carefully swing the housing toward the left to remove the push on connector. The wire should then be slipped through the cutout. If the wire is routed through the rear cutout on the left side (item 3, Photo 1), the housing can be lifted straight off.

At this point the main circuit board will be exposed. If your board looks like Photo 2, you have one of the older long boards. If it looks like Photo three, you have one of the newer short boards. Do not be concerned about this. The boards are functionally identical. Commodore is simply making use of some new large scale integrated circuit components to replace a large number of older small-scale integrated

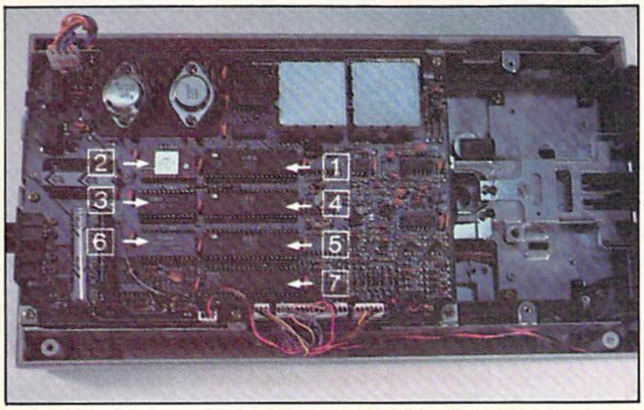
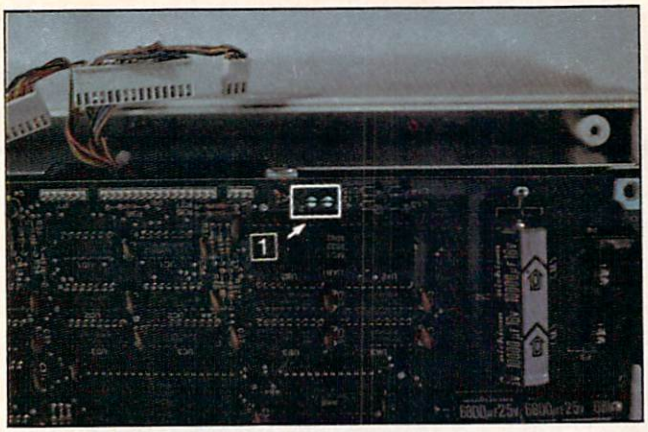


PHOTO 3: Short board (showing hardware rearrangement, utilizing large-scale integrated circuits).



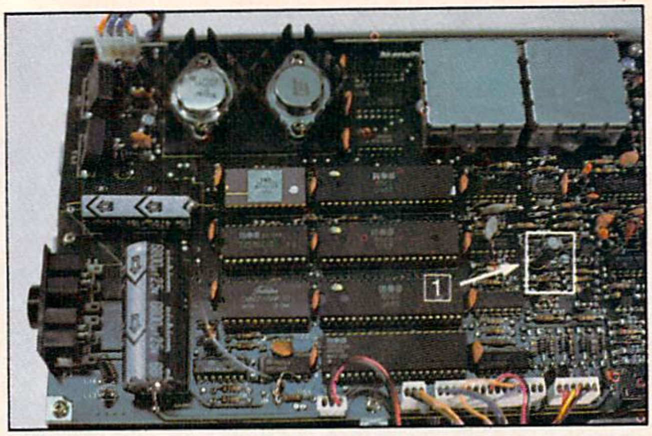


PHOTO 4 (left): Device number selection pads on long board; PHOTO 5 (right): ditto for short board.

circuits. The chip that does this is item 7, Photo 3.

Referring to Photos 2 or 3, item 1 is a 6502 microprocessor. This is actually the same device used in the VIC-20, Atari and Apple II computers. It is also very similar to the 6510 microprocessor in the Commodore 64. Items 3 and 4 are 8 kilobyte Read Only Memories (ROM). The actual instructions which make up the Commodore Disk Operating System (DOS) are stored here. It is items 1, 2 and 3 which make the 1541 an "intelligent" peripheral. This means that the DOS does not have to be loaded into memory every time you turn your computer on. It is also the reason that the Commodore DOS does not steal any memory from your computer's program space. Incidentally, if you have a 1541 disk drive, the ROM which will have to be replaced to upgrade it to a 1541 is item 2. Items 4 and 5 are both 6522 Versatile Interface Adapter (VIA) chips. These handle all the data transmission between the computer and the disk drive and between the components of the drive itself. Item 6 on Photo 3 is simply a 2 kilobyte Random Access Memory (RAM) chip which replaces several smaller RAM chips on Photo 2.

Don't go pulling any of these chips out of their sockets. Although modern semiconductors are quite sturdy, there is still some danger from static electricity during handling. As such, the chips are much safer in their sockets.

We have now reached the point where the drive device number can be changed. Refer to Photo 4, item 1 on the long board. You will notice four semicircular pads on the left side toward the back of the board. These are set up in pairs with a thin printed circuit trace connecting them. If the trace between the pads labeled 1 is broken, the device number will be changed to nine. If the trace between pads 2 is broken, the device number is changed to 10. If both traces are broken, the device number will be 11. The original device number can always be restored by simply soldering across the broken trace. Refer to Photo 5 for the short board. In this case the circular pads have been moved to the center of the board and they are no longer numbered. Nevertheless, the procedure for changing device numbers is the same.

Up to this point, all we have accomplished is actually sanctioned by Commodore as the procedure is briefly described on page 40 of the 1541 user's manual. As such these operations should not constitute a violation of the warranty. We will now venture into forbidden territory, so proceed with caution.

The next step is to remove the main circuit board to gain access to the drive mechanism, which will allow cleaning of the read/write head. You will notice a number of connectors around the periphery of the circuit board which will have to be removed. These are all slip-on connectors and should not present any problems. Take careful note of the orientation of each connector to avoid reversing them during reassembly. This should not be a problem in any event, as the twist of the wires tends to hold the connectors in their original orientation. To further simplify matters, each connector has a number one molded into one side of the nylon housing. This corresponds to the number one on the printed circuit board next to the connecting pins, although some of these markings may have been left off on the short boards.

A total of six connectors will have to be removed. Working with the long board (Photo 2), the power on LED connector (item C1) at the front left hand corner will have been taken care of when the metal housing was removed. On the short board (Photo 3), this connector is now the rearmost on the left side of the board. Be particularly careful when removing the black head connector (item C2) located at the right side toward the front of the long board (Photo 2) and in the front left hand corner of the short board (Photo 3). This connector carries the signals to the read/write head, and the wires are rather thin and fragile. Note that one of the pins on the black connector is missing, leaving only four out of five connections.

The four remaining connectors are at the same location on both boards. The main power connector (item C3) is at the back right hand corner of the

board. The drive mechanism interface connectors (items C4, C5, C6) are at the center left hand side in a group. There are a total of seven screws holding the main circuit board down to the main chassis. Three are along the top left side of the board at the front, center and back. Two more will be found on the top of the board at the front right hand corner and at the top center position at the back. All these locations have been marked as "S1" in Photo 2. The last two screws are on the right hand side (not on top) towards the back of the main chassis. These actually screw into the heat sink at the back left hand corner of the board. (The heat sink is the black metal contraption with the rectangular fins pointing straight up. This device dissipates the heat generated by the +5 volt and −12 volt voltage regulators which supply power to the drive electronics.) Once all seven screws and six connectors have been removed, the main circuit board can be removed and set aside.

At this point the entire drive mechanism will be exposed as shown in Photo 6. The main power transformer (item 1) is the large square device towards the back. The round motor (item 2) located at the rear left hand corner of the drive mechanism subchassis is what spins the disk during read and write operations. (The disk remains stationary at all other times.) Further disassembly of the drive is not recommended at this time, since the read/write head is now accessible for cleaning.

Opening the drive door will lift the pressure pad arm assembly off the head, preventing possible damage to the head if the pressure pad arm should accidentally be released while cleaning. The pressure pad assembly (item 3, Photo 6) is shown lifted out of the way in Photo 7. The white rectangular area with the bisecting black line is the read/write head. This head is encased in glass and should be treated with great respect. Carefully wipe the head with a fresh cotton swab dipped in isopropyl alcohol. Use the compressed air or squeeze blower to clean out any cotton strands. Avoid touching the head with any hard objects. At this point, the drive should be reassembled by reversing the order of the steps above. The following notes are for advanced tinkerers only!

For further disassembly, the entire drive chassis should be removed from the case. There are a total of six screws, located on the left and right sides of the chassis, which have to be removed. The drive mechanism subchassis is held in place with four screws located on the right and left sides of the main chassis, labeled S2 in Photo 6.

The pencil in Photo 9, which shows the bottom of the drive mechanism, points out an earlier mechanical fix to the drive mechanism. In this case the drive pulley with the radial strobe markings was a press fit on the drive shaft. The small screw in the center of the drive spindle pulley was added to pre-

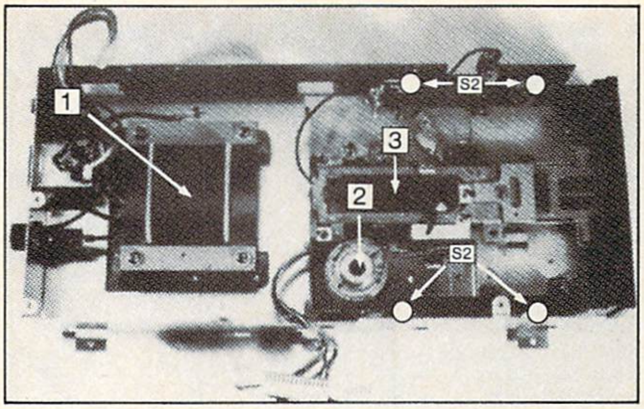


PHOTO 6: Main chassis.

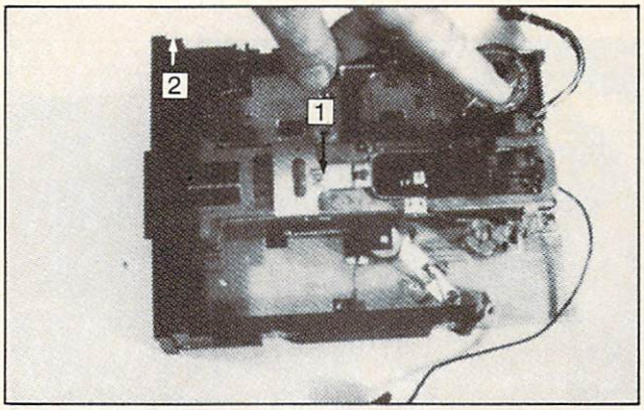


PHOTO 7: Read/write head.

vent it from slipping off the shaft. (This screw was missing in early production models of the 1540 disk drive.) A slipped pulley can be recognized by a loud scraping noise made by its hitting the metal chassis. This problem can be easily fixed by using a drop of cyanoacrilate (crazy) glue on the shaft when pushing the pulley back on. The pulley should be nearly all the way on the shaft before adding the glue, as it sets very fast. The drive door will have to be closed and the top of the spindle on a firm support when this repair is performed.

The following problem was reported in the newsletter of the Catalina Commodore Computer Club, 8037 East 18th Place, Tucson, Arizona. It concerns the drive hub pressure assembly which holds the disk against the spinning drive hub. This assembly is supposed to have slippage, but it occurs at an incorrect point. The assembly consists of a spring sandwiched between stainless steel thrust washers. This spring tends to wear away the washers causing an intermittent grinding, squealing, or howling noise. The design of this assembly seems to be at fault in that the individual components do not work well together.

Lubrication of these components seems to cure the problem, at least for the short term. The use of any kind of lubricant in the drive area is in general not recommended, and should be approached with extreme caution. We suggest that a small drop of

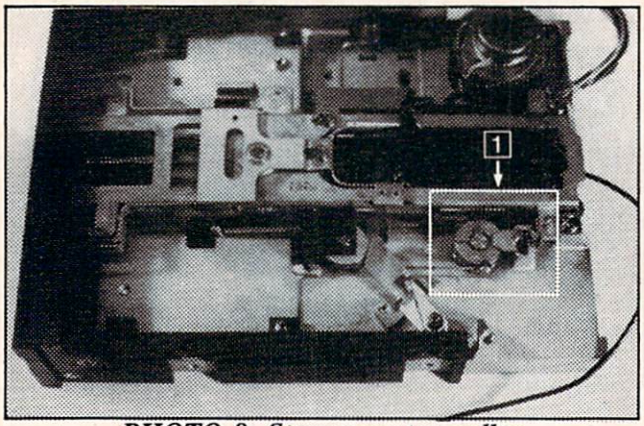


PHOTO 8: Stepper motor pulley.

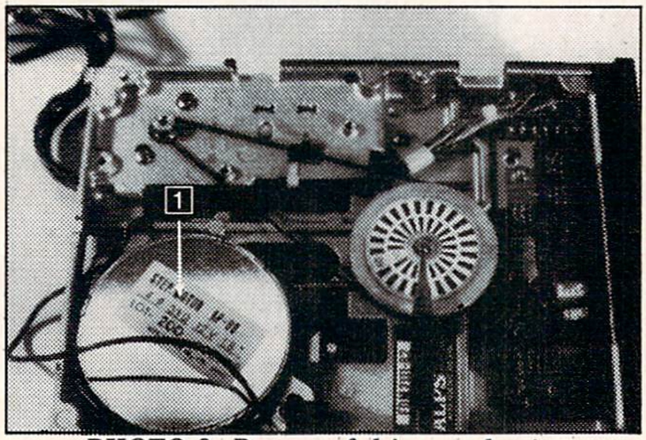


PHOTO 9: Bottom of drive mechanism.

number 5 light machine oil be applied to the tip of a jeweler's screwdriver. A measured amount of oil can thus be transferred into position. (This oil is generally supplied for lubrication of sewing machines and may be available in squeeze tubes with long applicator tips.) The components to be lubricated can be reached by removing the small "C-ring" (item 1, Photo 7) at the center of the door arm assembly. The drive mechanism door (item 2, Photo 7) will have to be removed to gain access to the components. The door can be moved only a limited amount because of the restriction caused by the wires to the red drive activity LED in the front panel. Use caution when performing this operation, as the two springs in the mechanism tend to cause the parts to fly apart.

We will close this discussion with a brief description of the head positioning mechanism. The stepper motor (item 1, Photo 9) is capable of rotating in 1.8 degree increments. This is converted to linear motion by a drive assembly consisting of a drive pulley, an idler pulley, and a stainless steel band connecting the two with the head assembly. This mechanism can move the head in increments of 1/100 of an inch. The drive pulley appears to be a press fit on the stepper motor shaft. It is thus possible that slippage occuring at this point may have been causing drive alignment problems as it is apparent that a misalignment of less than 1/200 of an inch can be serious. This conclusion is speculation of our part, as we do not have sufficient information regarding the problem at this time.

Many of you have probably wondered about the loud chattering noise which occurs when the drive formats a disk. This is caused by the stepper motor pulling the head to the outermost position on the disk. A protrusion on the drive pulley is brought against a fixed stop (item 1, Photo 8) at which time the stepper motor hits this stop several times to set the head in a known starting position. This is what causes the loud chattering noise. The quieter burping noise which you may hear on occasion is caused by the stepper motor moving the head to a different track on the drive surface. This tends to happen more frequently as the disk is filled, because of the way the DOS allocates disk space. The center tracks are filled first, moving radially outward. Thus considerable head movement occurs when the disk is nearly full, as the head shuttles between an inner and outer track. If a read or write error should occur, the stepper motor will pull the head out against the stop, causing the loud chattering noise. This is done to reset the head position.

Depending on the nature of the error, the drive may perform this operation several times before giving up and reporting an error. This chattering noise will occur with commercial software that uses "bad sectoring" to copy protect the disks. This is the practice where an unreadable sector is deliberately written on the disk surface at a known location. Since these bad sectors cannot be reproduced during a copy operation, they are used as a check for copied disks. This practice may actually be a contributing cause to many head alignment problems. The continuous chattering caused during attempts to copy these disks puts considerable strain on the drive components. As we mentioned above, a misalignment on the order of 1/200 of an inch can cause read problems. It is conceivable that the repeated impact of the pulley against the stop can cause the drive pulley to slip on the shaft.

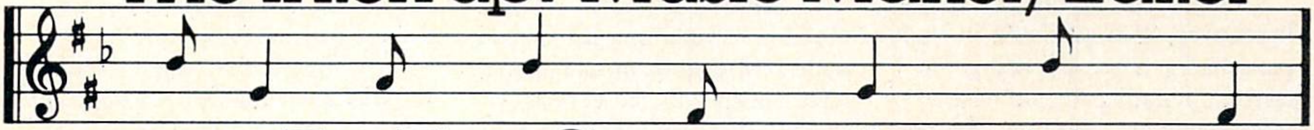
Some elementary precautions will extend the useful life of the disk drive. Maintain adequate clearance above the drive to allow dissipation of heat. Two or three inches is generally sufficient. Avoid smoky or dusty atmospheres. Smoke or dust particles can play havoc with the disk surface.

We have noted a minor bug in the disk operating system. This occurs if a disk is formatted immediately after a disk error occurs and results in an improperly formatted disk. The error should be cleared by reinitializing the drive with a properly formatted disk or performing an error free read or

Continued on page 78

By Pete Lobl

The Interrupt Music Maker/Editor



For The Commodore 64

PART II

This month Basically Machine Language takes a look at some sample music for you to enter into your music editor.

Before giving you the sample song, *Zocatera,* I'd like to address a nice bug that occurred in the program during the EXIT (*X) command. Retype these lines, then save the corrected version.

```
20207 FORI=1TOLEN(IN$):POKELO+I-1
,ASC(MID$(IN$,I,1)):NEXT
20300 PRINT"{CD}{CD}TAKE IT EASY,
 HOPE YOU HAD FUN !!{CD}":END
```

[1] is the name you gave voice 1. [2] is the name you gave voice 2. [3] is the name you gave the machine code (*X) or exit. [main] is the name of the program that will use music. Normal speed is 57.

To turn on music within the main program, enter:

```
SYS22976:POKE54296,15
```

15 is the volume POKE; it may not be needed. To turn off music:

```
SYS23034
```

Now here's the loader for the program:

```
5 ON A GOTO20,30,40
10 A=1: LOAD"[1]",8,1
20 A=2: LOAD"[2]",8,1
30 A=3: LOAD"[3]",8,1
40 POKE 56,89:CLR
50 POKE 56325,[SPEED]
60 LOAD "[MAIN]",8
```

Now when you exit (*X) the program, the machine code containing the interrupt will be saved to either disk or type as you specify. This way you can load the interrupt machine code and the music without the editor. In this procedure, you can mix the music with your existing BASIC or ML programs. Use this loader program to automatically load the ML and MUSIC for you. The loader program also lowers the top of memory for you; if the top of memory weren't lowered, the variables would overwrite your music, destroying it. Altogether you lose around 18K for the privilege of having this massive amount of music in the program at once. Most programs will still fit in the 20K left, but if for some reason your program doesn't fit you'll have to wait till next time when I'll discuss how to relocate the music and ML to a more convenient and economical location. In the meantime, keep experimenting and have fun. Try typing in a sample song—I think you'll enjoy it!

Remember, to enter an attack decay sequence (the first block), type *C; this will get you into the change mode. Next, type DSR and you will be able to enter attack-decay and sustain-release. On returning to the normal editor screen, you can then enter the music note by note till the end. Before entering the music you should be aware of one thing: you will have to turn off the auto separation feature. To do this, enter the change command, *C, then give the AV suboption. When the computer asks for the new duration, enter 0. Now you should return to the change menu. Enter X to get back to editor. Whew! Now you can enter the music for Voice 1. After finishing the music for Voice 1, follow the same procedure for Voice 2. To get into Voice 2, enter the change command (*C), then type V for voice change at the subcommand input. Now type X for exit and you will see Voice 2 in the upper right corner of the editor screen. Now follow the

Continued on page 70

BOOK REVIEWS

WRITING IN THE COMPUTER AGE ***by Andrew Fluegelman & Jeremy Joan Hewes***

This Doubleday book ($19.95 hardcover; also available in trade paperback) is must reading for anyone planning to purchase a microcomputer, and who intends to do a significant amount of word processing. Taking the subject of writing on a computer one step at a time, the authors guide the reader through the jargon jungle that has been inflicted on the world by techies who, like the alchemists and sorcerors of old, have attempted to create their own secret language. *Writing in the Computer Age* tells you how to choose a computer and a word processing program which will fulfill your own needs. Best of all, even the completely non-technical person will benefit from this book.

In addition to being written in a clear, straightforward manner, authors Fluegelman and Hewes have added another very nice touch. They describe in detail how they used their computers to write the book, communicate with one another via telephone and modem, and finally to transmit the edited manuscript to the typesetter. The book is worth every penny you pay for it, and whether you are on the brink of going electronic, or have already taken the plunge, you will be glad you bought it.

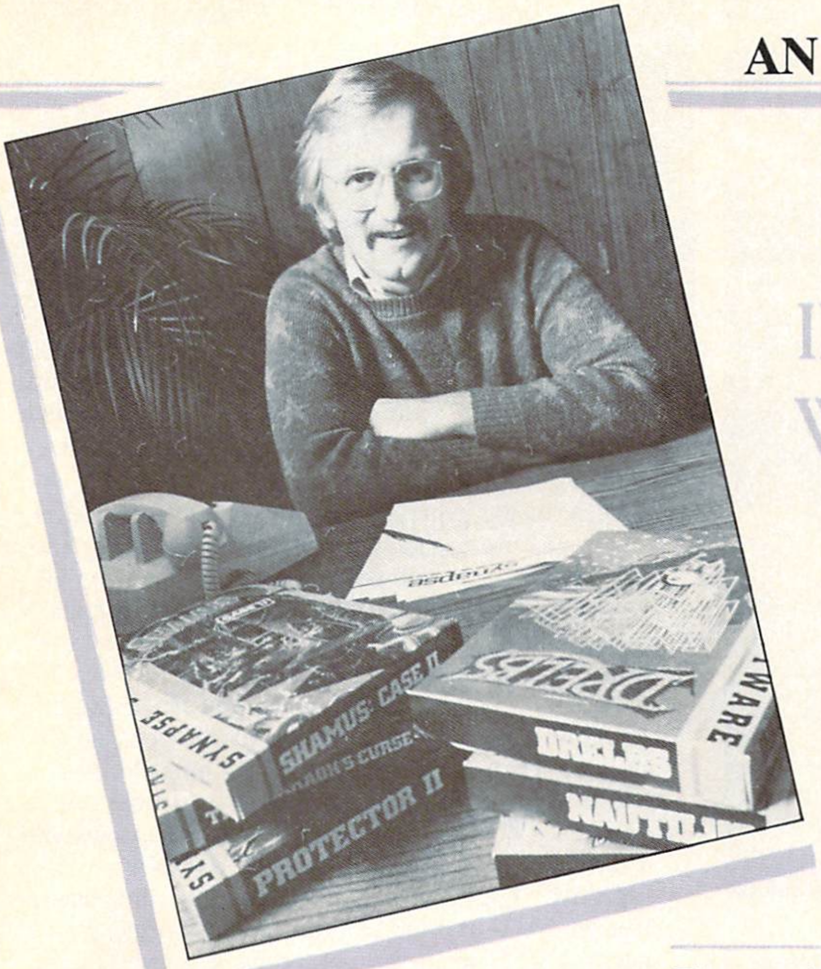
While on the subject of books about computers, a brief mention should be made of the two Peter McWilliams Books, *The Word Processing Book,* and *The Personal Computer Book.* Among the first books written in non-technical language, the McWilliams duo taken singly or together are worth reading. He approaches the subject with a kind of zany humor, which tends to humanize it, and that is good, if his cutesiness doesn't become too cloying for your taste after awhile. The author also provides a fairly comprehensive buying guide, which, despite the fact that such an effort becomes obsolete before a book even goes to press, is still of value, because it offers practical information that can still be used.

ELECTRONIC LIFE, How to think about computers ***by Michael Crichton***

This unusual book (Alfred A. Knopf, $12.95) by the author of such novels as *The Andromeda Strain, The Terminal Man,* and *the Great Train Robbery,* is truly nifty. It is not only for the potential buyer and the computer novice, but for the "computer widow," and everyone in creation who has ever been bored stiff at some gathering by one or more computerniks speaking in electronic tongues and preaching the gospel of the chip and the byte.

Although Crichton goes over some of the territory that has been covered by others in explaining the anatomy of the computer and how to buy one, he does it with intelligence, and in thoroughly human terms. Consequently, by the time you finish the section in which he explains some of the more obtuse computer jargon, you know that when you encounter it again, you will not be intimidated or put off.

—Bernhardt Hurwood



AN INTERVIEW WITH

IHOR WOLOSENKO

The president of Synapse software talks about his company's new line of textual adventure games, language and experience, game theory and the self-awareness of a shoe salesman.

By Tim Moriarty

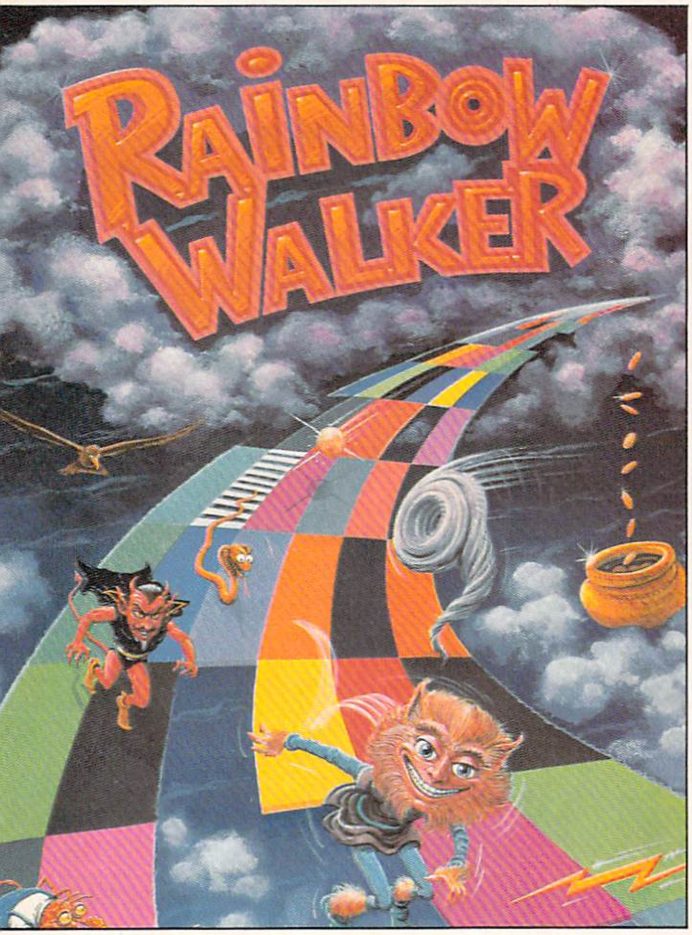
AHOY!: We understand that you have a background in physical therapy. How did you make the jump to computers and game design?

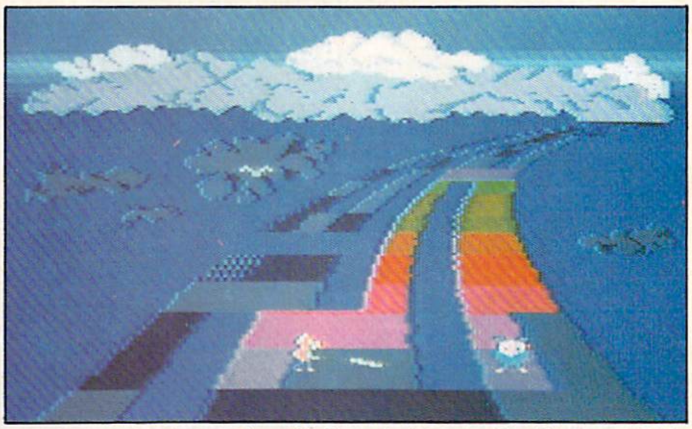
IW: Before I became interested in computers I was teaching indirect methods for physical therapists. Before that I was photographer in Boston for about ten years. Then I moved to Berkeley, where I got very interested in what the active elements were in the Tibetan language and how those elements influenced the language; how you extract active cultural elements and translate them into language.

That led me into the study of hypnosis and therapy. I studied with a couple of guys in Santa Cruz for a while who were interested in linguistics—how people derive meaning out of their experiences, how they represent it in language, and how language shapes experience and how experience shapes language.

Then I wanted to formalize my thinking. The computer seemed like a real good way of doing it. I studied subsets of natural language, machine language, BASIC, Fortran, and the other languages that were structured on the computer. I wanted to make some sort of analogy that would translate to the human experience. It served as a very good model, a simplistic one, but it led me into more precise ways of thinking about what experience is and at what level you can change people's experience.

I was interested in games, in emotional reactions to games, and the differences between a reader and a protagonist in novels. I was interested in how a computer structures the environment for someone to react with. That's really the direction that I'm most interested in right now. Who are you? Who is the character you're representing? Does your character have attributes in the same way you have attributes?





Synapse plays a fanciful variation on the **Enduro/Pole Position** *type of game with* **Rainbow Walker.**

Does your character have memory? What things do you expect the player as the player to remember, what do you expect the player as a character to remember?

AHOY!: One of your company spokesmen referred to the psychological "hooks" that you like to introduce into your games.

IW: I'll tell you a little about hooks. I was interested in the player and the projection of the player through the character in any kind of game, whether it's an adventure game or a skill-and-action game. One of the things I've experimented with is giving the projection of the player some volition in addition to merely being controlled by the joystick.

For example, if you take a three second pause while playing *Alley Cat,* the cat kind of takes off on his own. He starts meowing and moving around by himself. You have to constantly be aware. Through the pauses you become aware that the cat has a personality of his own, rather than you feeling that you have an inanimate thing that you have to animate through some force of will. It's a cooperative effort. Obviously the proportions of that can change. We're experimenting with a teamwork approach.

Another thing I'm experimenting with is self-adjusting games. I'm very interested in the whole concept of the game being responsive to the level at which you are engaging it. In order to be really successful in games, you have to engage players on all levels of expertise. The way that games have traditionally done this is to present a learning curve. When you first enter the game you say, "This is trivial, no problem at all." When you first engage *Pac-Man* you say, "Well, I can get the dots, no sweat." Then, as the area becomes more limited and the ghosts seek you out, it becomes more difficult very quickly. Unfortunately, what happens with a linear set-up like this, each game becomes repetitive. You have to work through the problems that you've solved already. It becomes boring. So the question is, how do you get the game to engage you at the level you want to play?

I don't want to level-orient games at all. I think that's an artificial distinction. When you play Monopoly, you never play it on levels.

AHOY!: The initial stages of the game change every time you set up the game board.

IW: Exactly, and they change on the basis of who is playing against you. So, how do you make a level-less computergame that is engaging, that has some way of rewarding the better player and yet brings along the player that isn't that experienced? The way we've done it is to divide the game into episodes. For example, *Sentinel* is rated according to how quickly the episode takes place, what the number of hits was versus the number of shots fired, what the basic tactics are, how much movement there was of the joystick, at what level did the ship come in toward you, what speed was their rapidity of fire, what other obstacles were there? There are many other factors built into the equation. Based on these factors, we change the kind of objects that come toward you, what kinds of things they do in reference to how quick they are, what their patterns are, how many engagements you have. Every minute that you play, the game makes intrinsic judgements

that gear the game to level of play.

In another game, we present a firing pattern that shoots one bolt at you, then backs off and circles around you. As you build more and more intelligence into the attacker, there will be patterns that he goes through based on what your reactions are. The game notices if you try to center in a particular way and it will call up a pattern based on what your strategy has been thus far.

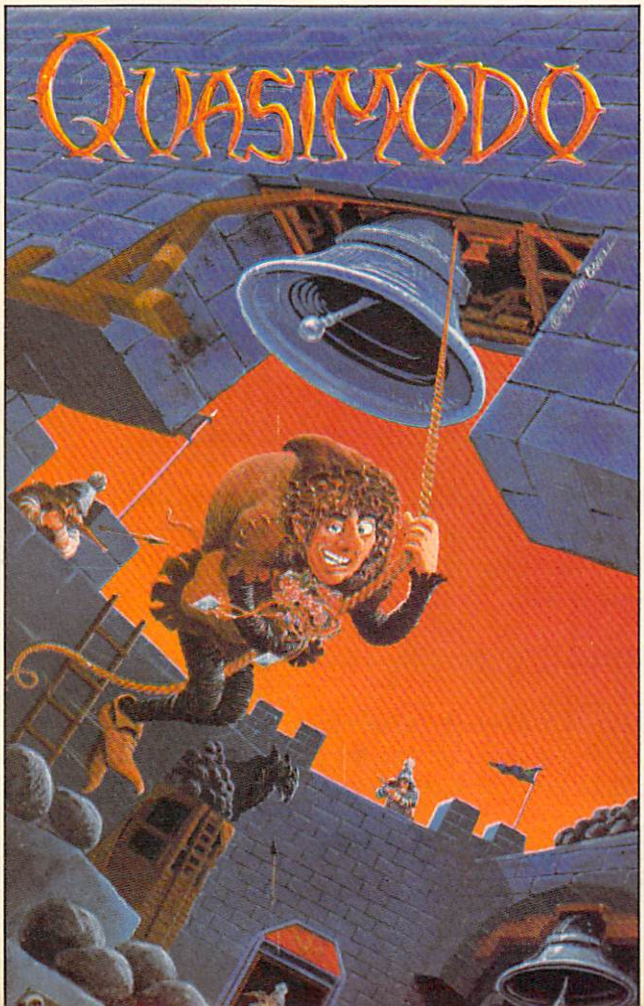
AHOY!: You can use that then to stay a step ahead?

IW: Sure you can. And that's an intuitive decision; it's not something explicit.

AHOY!: You can bluff the computer.

IW: Right. If you're rated on the basis of every minute, how can you plan a strategy that will fool the computer into shooting where your character is not. This is where I think that computergames begin to diverge from videogames: the fact that they have this artificial intelligence, that they can rate how you're doing and can interact with you on an individual basis, rather than establishing a pattern and expecting you to adhere to that pattern

AHOY!: And that is your answer to the problem of maintaining the interest of players of unequal ability or experience?



IW: Our answer is: if all of a sudden the enemy ships take two or three hits to kill because their shields come on just before you fire because they're anticipating your pattern. . . then you will get more points for the kill. Each game you play is completely different. This is the first step in determining how to make the computer react to you as an individual.

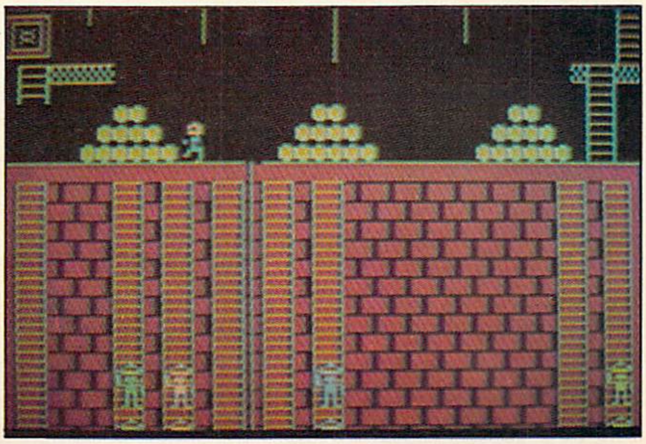
AHOY!: You said before that the response is intuitive on the part of the player, not explicit. Could you expand on that?

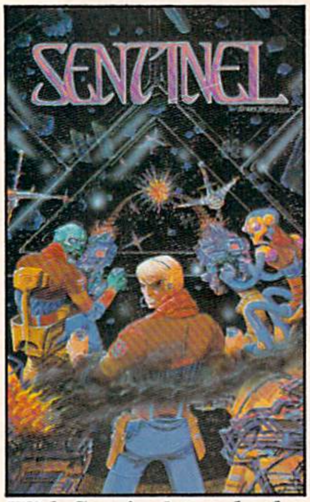
IW: When you're talking about the building blocks of games, you have to engage people aurally and visually. I don't care if people intellectually understand what's going on or not. The most effective learning in anything that we do comes on a subliminal basis. Every time you hear that warning siren, a certain kind of emotional response is evoked. The designer must take all sorts of elements into account and really think about them. The designer must ask himself: is there an arbitrary association here, or is this sound really associated with this action each time? Is there some kind of subliminal learning and response matrix being set up as you're designing the game all the way through? If the game looks good, that's fine. That'll engage some people who are visually oriented. However, if the game doesn't have playability, it will ultimately fail, and it will disappoint the people that it impressed in the first place. In fact, it will disappoint them more than if it didn't promise them as much in the first place.

AHOY!: You use humor to elicit an emotional response in the player. What other elements are employed?

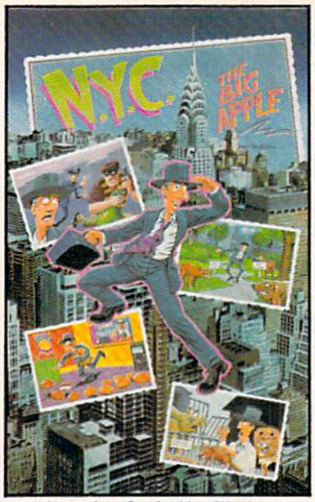
IW: Well, for one thing, I feel that to the greatest extent possible, the control of the game should be intuitively obvious. The game should not have a cumbersome overlay. For instance, if, in order to

A light-hearted game based on a classic of tragic proportions: help **Quasimodo** *defend his tower.*











With **Sentinel,** *each player will find his own appropriate level of skill.* **Shamus 2** *is the sequel to* **Shamus,** *the best-selling shoot-'em-up that dispatches a futuristic detective on a room-by-room search for the mysterious Shadow. This time a map has been provided.* **N.Y.C.** *and* **Slamball** *are brand new offerings.*

move left, I really have to move left to point myself there, and then I have to go forward to engage my speed and then go back again—that's a difficult sequence of events to engage the player with. If you pull the joystick back with the button pressed, you should notice what happens and say, "Okay, yeah, that's what should happen."

AHOY!: The Commodore 64 employs Atari-type joysticks. Do you wish you could design to more complex controllers, such as the Intellivision controllers?

IW: I don't know if I like that type of design because, while it is a nice package and it gives you multiple functions, it does require that one hand has to make discriminations that might not be made quickly enough in real time.

In the coin-op *Discs of Tron,* there is one control that directs where your discs are to go and another one that controls the player. That's a very nicely designed game. It has taken multiple controllers and made them integral to one another.

AHOY!: You're saying then that overly-complex controls can interrupt the player's emotional involvement in a game?

IW: You wouldn't turn on the lights in the middle of a movie, because that is going to reorient the moviegoer to the audience around him. In game design, you have to look at everything that is going on in very precise detail and understand what part of that process is important to the game, and what part interrupts. If you keep on forcing people to do something that they don't want to do, the game is not going to be successful. Oh, it may be marginally successful, but there will be complaints about it, some not fully verbalized.

AHOY!: Complete involvement is obviously very important to you.

IW: Whether the game is skill-and-action or whatever, the object is to create an experience that is so totally engrossing that it provides a way for people to experience their lives differently.

I'm now designing a game called *Enigma.* You find yourself on a desert island. You're face down in the sand. You don't know who you are or why you're there. It's a game of discovery.

AHOY!: A completely textual game?

IW: Yes. I'm trying to incorporate a totally interactive environment. For example, in that situation, some player/person might go swimming, another might roll over and cry out in anguish, another might dig in the sand, another might climb a tree to gain a vantage point and see where he is. On the basis of what you do, objects will exist and situations will evolve. Whoever plays the game will have completely different experiences.

AHOY!: Will there be any dead-ends?

IW: No, not at all. There will be a structure that will vector the person toward things to happen, meeting characters and so forth, that lead to a conclusion.

AHOY!: To the same conclusion, ultimately?

IW: No, There is a possibility of many conclusions. This is a game of discovery. It is not a puzzle or a riddle that you have to solve in only one way.

AHOY!: Back to the unfortunate wretch in the sand: is that always going to be the same person at the

conclusion? Say, a shoe salesman from Philadelphia, whether he panics or digs or swims or whatever?

IW: Sure, but I think that the fact that he's labeled a shoe salesman is almost irrelevant. It's the experience of discovering that you're a shoe salesman and the various ways you can do that. For instance, I could remember that I'm a shoe salesman by someone handing me a note that reads: "You're a shoe salesman." Or I could discover a shoe and remember, or even discover something fascinating in a footprint; I would say, "I think I recognize that but I don't know why. It seems appealing." So, ultimately, the question is: what is compelling about experience? What is it that is compelling about being a shoe salesman? Perhaps the fact that you discover who you are is irrelevant to the game. The experiences that are presented to that end are the important thing.

The human mind tends to try to create patterns and to inject meaning where there may not be meaning. But with the addition of experience, new information, a different interpretation is possible, a whole new interpretation. So how do you manipulate that creation of meaning, since there is no ultimate resolution?

AHOY!: Is there an enormous following for textual adventures?

IW: We're a visually-oriented society. Until we can engage people with language and interpreting language, these games are not going to achieve tremendous popularity. You have to make certain assumptions about your audience, and the audience that enjoys reading is a relatively small one. However, it happens to be fairly congruent to the audience that has computers, so we're lucky in that sense.

But I do feel that it is going to allow us to exploit the maximum power of the computer. The problem with computer graphics is twofold: one, they are an impoverished version of what you can create for yourself; and two, they take up alot of memory that you can use for artificial intelligence, and having the game play you. I'm more interested in having the game play you than you playing the game.

AHOY!: What are you calling your text adventures?

IW: We're calling them Microworlds.

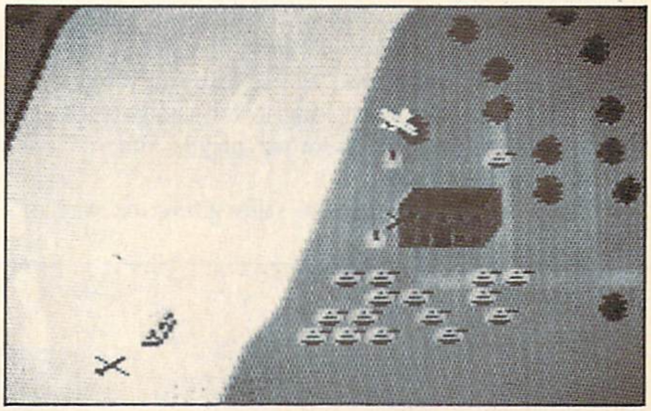
AHOY!: And you have novelists at work on them?

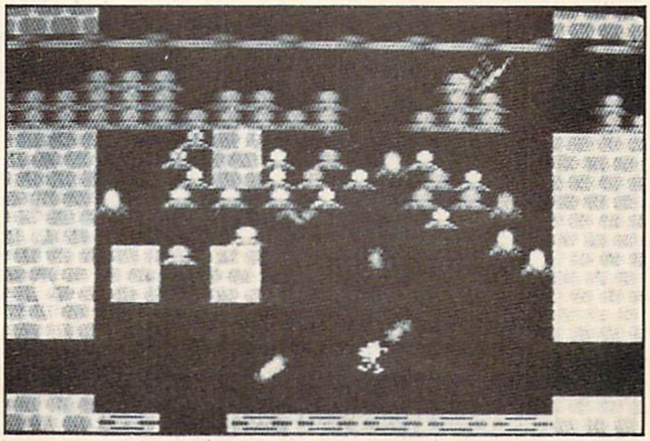
IW: In fact we do, but I can't tell you who they are because we haven't signed all of the contracts. Top, top novelists

The problem with these adventure games thus far, even the more interactive ones, is that you have the feeling of being in a corral. You go this way and someone says, "You can't go that way." If I say "Toss something" and it says, "I don't understand that word" when it just used that word in a description, it drives me up the wall. It totally stops the experience for me. We're going to have to work with some of those obstructions until we can solve some of the problems: not processing time, just putting the computer's power to better use.

We're addressing the problem with our cocktail party game. It has hardly any physical space at all. Rather, you go from person to person. You overhear things, you say certain things and later hear variations of what you said come back. Characters remember certain things, forget others.

The most intricate puzzle is not a Rubic's Cube, it's a person. And it's a character that changes. When you read bad fiction, the character comes in, he interacts with alot of people, and he goes out exactly the way he came in. When you read a Tolstoy novel, the character is totally different at the end of the novel than when he came in at the beginning. And that's what we're trying to do. There is no reason why you have to be the same person during a game either. You could have a changeling-type game, where you're a person at one point, you're a dog at another, a bat at another.





***Aerial combat with* Blue Max *(top) and sleuth-and-shoot in pursuit of the Shadow in* Shamus II.**

AHOY!: Or you could begin the game placid and logical, and by the end you're a crazed murderer.

IW: Exactly! Then your responses would change and so would the responses of the characters you interact with. If you kick someone you probably can't expect too much help from them.

AHOY!: One of the first videogames to attract female players on a wide scale was *Pac-Man.* Why do you suppose that was, and why do you think women are less involved with video and computergames than men . . . generally speaking of course?

IW: Looking back, you can interpret something in any number of ways. You can find a pattern in *Pac-Man* that was cosmically introduced or that, synchronistically, society was ready for. You can do anything you want. Once things are done, you can say "Ah yes. The pattern is obvious."

I don't think it was specifically *Pac-Man,* assuming that you are correct. I think it was the learning curve. In that game, there's no ambiguity. It seems like a trivial task, something that is easily done. There's instantly alot of control. It's no big deal. Whereas in other games, the decision pace may be too fast. Someone who has never been involved in that particular game says, "That game is too fast. I can't learn it."

So, I don't think it spells a difference between men and women but between experienced players, who are willing to take that step, and inexperienced players. I don't think you can sexually differentiate other than that women belong to the category of videogame players that have not played alot.

AHOY!: Are computergame sales going the way of the videogame business: smash or dud?

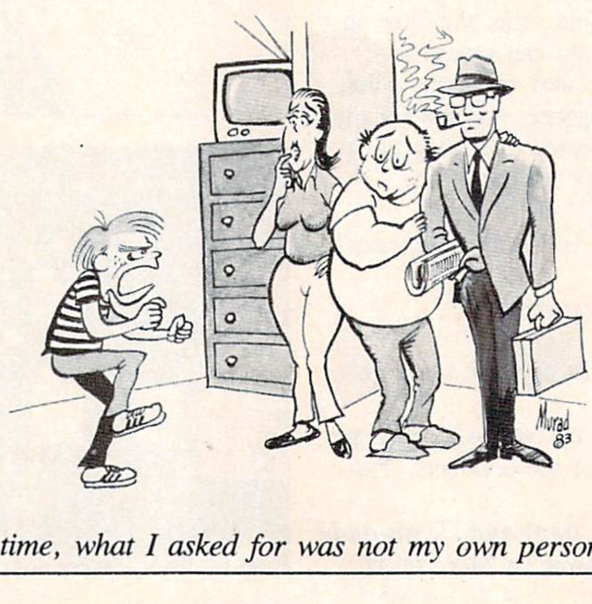
IW: The analogy I make is that computergame companies are equivalent to the recording artist. For example, Elton John is the equivalent of Synapse, rather than a particular programmer. I think people are relying more and more on companies to develop good games, because there is such a proliferation of garbage.

Hit or dog? Yeah. Success really depends on how well you can understand what people want at a particular time. It's all timing. You must be able to anticipate where the world is going to be and match it. And the trick is matching it and leading it to something new. You match a person's expectations and say, "Okay, I got you. Now let me show you something else."

AHOY!: Would you care to speculate on future game technologies such as laserdisc players interfaced with home computers?

IW: Laserdiscs would have to be used completely differently from the way they are used now to satisfy me. It all has to do with how quickly you can branch, and how many branches you can generate at one time. Until laserdiscs are used as a mass storage medium with much faster processors, there is going to be very limited interaction. There are prescribed branches, and that's boring. You're committed to a particular course of action. When they play *Dragon's Lair,* many people don't even know whether they've moved the joystick correctly until they've sunk about twenty five dollars into it.

New styles are beginning to emerge, and they're not "Move the cursor, shoot the dot." I think that games are coming out of the fingerpainting stage. We're only now developing brushes. □



"For the last time, what I asked for was not my own personal commuter!"

Stop Gambling. Start Winning. Now.

It's a fact. You will beat the dealer if you play Blackjack correctly. In Las Vegas. In Atlantic City. In dozens of foreign countries throughout the world.

They haven't changed the rules. Even multiple-deck games pose no problem if you play properly. You can win just as easily in 1984 as you could in 1961 when the first Blackjack strategies were created.

This ad is your cue to join the small group of Blackjack players who are no longer gambling. Become a strategy player and win. Consistently.

The Obstacle

Despite the wild claims made by the Blackjack system charlatans, it is not possible to learn an effective strategy overnight. Learning an effective strategy takes time and discipline. If learning a strategy were easy, everyone would be making a living playing Blackjack. As it stands, less than one percent play well enough to make money.

The Solution

BLACKJACK TEACHER simulates, in precise detail, the events that transpire in actual casino play. The display screen depicts the top view of a Blackjack table. You interact with the program just as you would an actual game. Computer controlled players occupy adjacent seats. All events occur in real-time.

BLACKJACK TEACHER teaches seven different strategies of varying complexity and accuracy. This spectrum of strategies allows you to select a strategy that suits your needs.

BLACKJACK TEACHER monitors your betting and strategy decisions (hit/stand/double/split/insurance). If your decisions are incorrect within the guidelines of your strategy, the system will display error messages showing you the correct decisions.

BLACKJACK TEACHER is the result of over ten years of Blackjack research. The strategies encompassed by the system were developed using computers. The more complex strategies are among the most powerful ever devised.

Complete documentation is included which tells you everything you need to know to become an expert strategy player.

The SOTA Story

SOTA Enterprises has consistently produced nothing less than the highest quality software. When you buy software from SOTA, we do our utmost to make sure you get your money's worth.

ATTENTION VIC 20 USERS

A new version of BLACKJACK TEACHER is now available for the VIC 20. Although not as comprehensive as the original 32K program, the VIC 20 version does teach Basic Strategy – a must for the Blackjack strategy beginner!

FILL OUT AND MAIL TODAY!

Name ______________________

Address ______________________

City ______________________

State ____________ **Zip** ____________

Make Check or Money Order Payable to:

SOTA Enterprises, Inc.
833 Garfield Ave, Suite 101
South Pasadena, CA 91030

Check Box

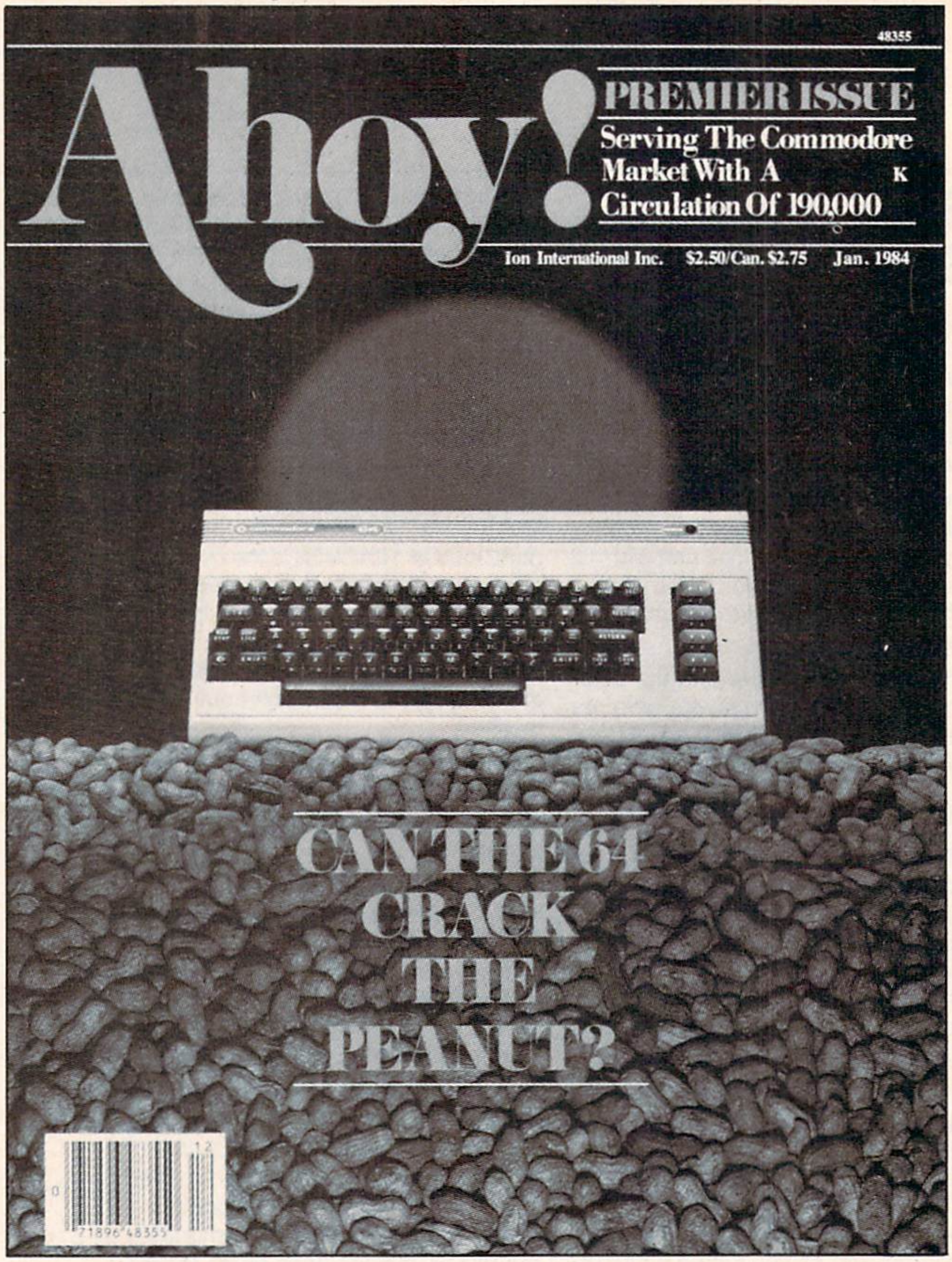
☐ VIC 20	($19.95)
☐ COMMODORE 64	($59.95)
☐ PET (32K)	($59.95)
☐ PET 2001 (32K)	($59.95)
☐ CBM 4032	($59.95)

Media

☐ Cassette ☐ Diskette

Postage Paid • California Residents add 6½% Sales Tax

Reader Service No. 46



HAS EVERYTHING!

SUBSCRIBE TO AHOY!

☐ Twelve Issues for $19.95 ($23.95 Canada and elsewhere)

☐ Twenty-four Issues for $37.95 ($45.95 Canada and elsewhere)

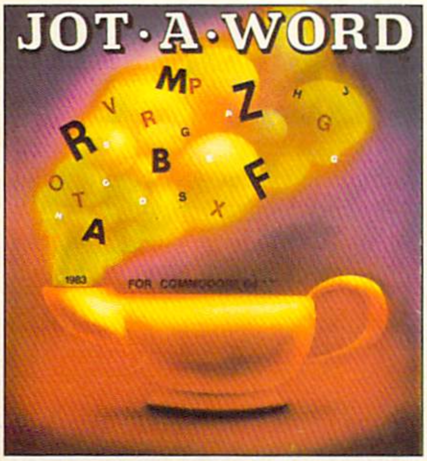
Name____

Address______________________________________

City____________ State____________ Zip____________

Send coupon or facsimile to:
ION INTERNATIONAL INC.
45 West 34th Street, Room 407, New York, NY 10001

REVIEWS



JOT-A-WORD
Micro-Ware Dist., Inc.
C-64, VIC-20
Disk; keyboard

This is a software adaptation of a traditional word game. The user selects a word (the computer admonishes the user not to let it see the word) and the computer also selects a word. With the player guessing first, computer and player take turns in guessing letters which the other's word might contain. The object is obviously to correctly guess the computer's word before it guesses yours. The program contains five thousand words (to which you can add more), and there are four levels of difficulty.

My 64 was quite good at this game; it took me a while to be able to beat it on the first level. As we see-sawed victory and defeat, I decided to choose the human option of cheating. Each time my 64 guessed a letter, I told it that its guess was incorrect. About the fifth time, it told me my responses were not, say, kosher. I relented, saying that I had made a mistake on previous turns. It pointed out that I was contradicting myself, but, like the good sport that I know it to be, gave up—"I'll give you the benefit of the doubt," it said.

If you enjoy word games or want to increase your word power, *Jot-A-Word* is an enjoyable option—and a worthwhile activity for the family.

—Steve Springer





ZEPPELIN RESCUE
Computer Software
C-64
Disk, cassette; joystick

Sometime in the future (we presume), Los Angeles is on the verge of an environmental disaster. The citizens have been advised to cluster on the rooftops of the buildings, there to await rescue. You have commandeered a fleet of zeppelins. You must fly over the city, scoop up the populace, and deposit them in the environmental protection centers.

There are two versions of gameplay. In the first (or practice) mode, the sky is clear, free of obstacles. In the normal mode, the sky is filled with acid rain-encumbered clouds. Should your zeppelin touch a cloud, you will explode, crash, and kill everyone aboard. (Crashing into a cloud: now that's an environmental disaster!) Also, brushing one of the city structures spells doom for your gasbag.

If you are looking for a fast action game, you will not find it in *Zeppelin Rescue,* for, true to life, the ship is ponderous and methodical. But as far as suspense and entertainment go, this game is a winner. While the laid-back pace may be frustrating at first, the game becomes involving and entertaining as the game progresses and as the required skills are learned. Tactics as well as coordination will be needed to maneuver around the clouds and skyscrapers, and to keep your zep refueled.

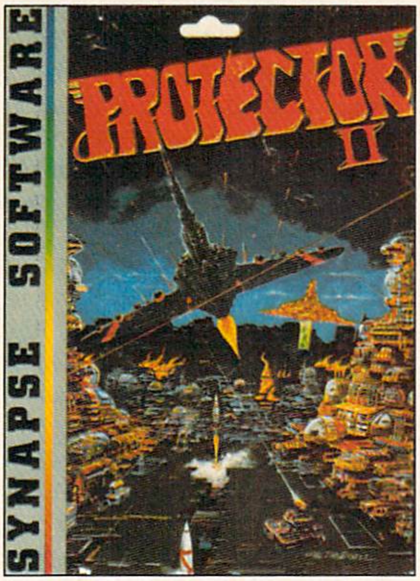
Surely there are computergame fans who are turned off by the hell-for-leather pace of most games or would prefer rescuing people to blowing them away. Since the game activity is limited, only the buyer can determine if it justifies a purchase.

—R.J. Michaels

PROTECTOR II
Synapse Software
C-64
Disk, cassette; joystick

The Slimehordes! Why, the very name of the enemies in this game makes one want to rescue the poor people from their gruesome grip.

The object here is to rescue eighteen people who are marooned in a city that is under attack from the Slimehordes. One by one, you must pick up the unfortunates with your needlefighter and transport them to the City of New Hope... all the while avoiding and blasting Slimehorde Chompers, Meteoroids and Rocket Bases. You must also keep an eye on the Fraxullan Mother Ship, which busily beams



up the people and drops them into an active volcano.

Once all of the city dwellers are successfully transported to New Hope, the volcano (Dragonmaw) erupts and begins to flood the city with lava. Once again, you must air-lift the citizens past the laser fields of Straak to the Verdann fortress.

The screen scrolls horizontally. The gamefield is large enough to attract continuous interest but not so large as to boggle the player's mind. What will boggle the player is the busy nature of the game: dodging enemies, rescuing helpless folk, and blasting insidious Slime-hordes. Not much strategy involved, but plenty of reflex, and the graphics are not slimy at all—in fact, they're very pretty.

—Robert J. Sodaro

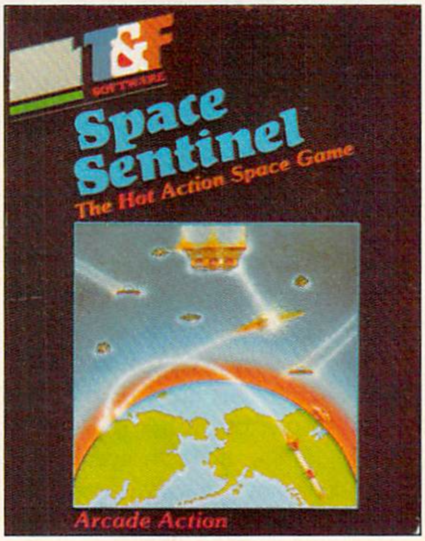
SPACE SENTINEL

T&F Software
C-64
Disk; joystick

In the universe of computergames, space is hardly a vacuum; it is filled to overflowing with hostile aliens and gallant defenders of mankind.

In this offering from T&F Software, your spaceship is in orbit around the earth. In early stages of the game, aliens make reconnaissance passes at the sphere, but as the action progresses, they begin to drop continent-obliterating bombs on its helpless denizens. Your job is to blast them, that job becoming more difficult as the game wears on.

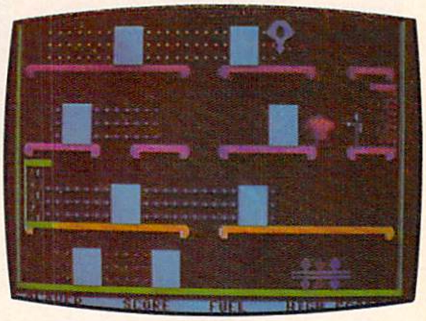
Uniquely, the screen in *Space Sentinel* does not scroll, nor is there any vertical or horizontal movement within the scene presented...nor does the player have any control over his or her space ship. An orbit is an orbit is an orbit; you endlessly circle the globe while your enemies approach and fall away. The one skill you must evolve is the ability to aim and time your fire; this is not an easy task, since you are moving and the aliens are moving. While you are honing that skill, you will watch whole continents on the planet's surface disappear under



the alien barrage. Once you have mastered the skill, however, you will watch your interest in the game disappear.

This is the best offering from T&F Software that we have seen, but their previous efforts, *Candy Bandit* and *Speed Racer,* were sub-standard. The gameplay in *Space Sentinel* is involving for a while, the graphics are an improvement—but T&F is taking simplicity about as far as it will go, and beyond.

—R.J. Michaels



MOTHERSHIP

Softsync, Inc.
C-64
Disk; joystick

With a few twists and touches, this game's first level is a *Star Strike* (or *Star Wars*/DeathStar trench) type of game; you are screaming down a corridor in space as alien ships attack you head-on, or swoop down on you from above.

Most games of this type offer either a cockpit view (total point of view, your ship is not visible) or the p.o.v. is from slightly above the ship you are controlling. In *Mothership,* your view is parallel with that of the ship's. This can serve to confuse you, obscure your view of the enemy, and interfere with your fire. Also, you are not shooting lasers, but rather a plasma torpedo which is slow and hard to aim.

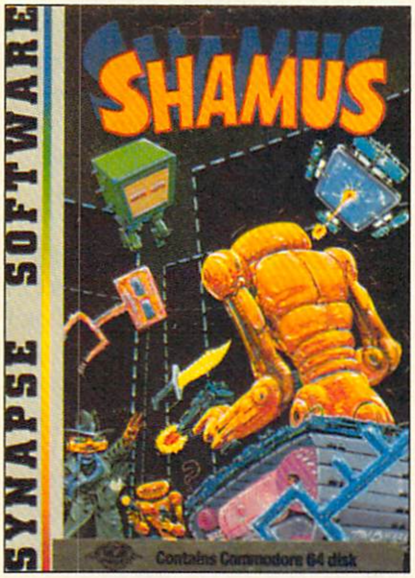
After dealing with the armada, you will progress to the second level, where you fight your way to the control room of the Mothership (in a multilevel climb-and-shoot scenario) and attempt to take command of her. Once you are in command, you return to the

Mothership's home base and destroy it.

The graphics in this game are quite good, with depth and imagination. The gameplay is highly varied—more like three games in one. And be warned: each level is very difficult to master.

Mothership offers a real challenge in each of its chapters (separated by an annoying return to the menu each time—a minor irritant) and an overall narrative and goal. Recommended for the player who is looking for a challenge, but be prepared to spend some time with it: this game is a real mother.

—R.J. Michaels



SHAMUS
Synapse Software
C-64
Disk, cassette; joystick

It is the 21st century, and you are in search of the insidious Shadow. You have managed to penetrate his lair, but now you must hunt him down through the 32 rooms on each of the four levels while defending yourself from whirling Drones, Robo-Droids, Snap-Jumpers and, of course, the Shadow himself.

The Shadow's lair is essentially a maze, with each room and connecting corridors revealed one portion at a time. Not only are there three levels of play, but you have a choice of three characters to portray: Marlowe, Bond, or Clouseau. The layout of the maze changes with each new character. The object of the game is to penetrate to the deepest core of the lair and to discover the Shadow's "secret."

Shamus is good company, which is just as well because discovering the Shadow's secret is a long and nerve-frazzling haul. Each time you enter a room you will be assaulted by too-many of the Shadow's minions, each scrambling about and trying to make you one with the universe (by blowing your brains out). Even if you clear a room of enemies, should you return to that room, you will be greeted by more of them. You must be careful not to touch the walls (zap! one life gone). Occasionally the Shadow will attack you; with several well-placed salvos you can stun (not kill) him, but you must be careful not to touch him (zap! another life forfeit). While you are engaged in combat with the Shadow and his henchmen, you must look for "clues" (question marks which can help or hinder you—you never know which) and "keys" (which will propel you deeper into the maze). All this you must do while keeping track of your position in the maze. Busy, busy.

Shamus, long a best-seller in other computer versions, is realized with Synapse's customary wit. Your Shamus pads about the maze on happy feet, despite all the chaos. Musical flourishes, such as the ongoing strains of "As Time Goes By," accompany the action. The graphics are flicker-free and colorful.

The directions recommend that you play *Shamus* with a friend, so that (s)he will be able to keep track of your location in the lair. We echo that recommendation. Still, if you can't find a game-loving friend (or don't have one), don't let that stop you: *Shamus* is a thoroughly enjoyable game with all the action and suspense that both novices and sophisticated gamers will demand.

—Robert J. Sodaro



SAVE NEW YORK
Creative Software
C-64
Cartridge; joystick

We don't know for a fact that Joe Jetson, designer of this game, is any more interested than Jimmy Carter was in saving New York. But Tokyo had been destroyed so many times by Mothra and Rhodan and Ghidra that when it came time to design a game in which winged monsters would devastate a city, Creative Software might have deemed a change of locale necessary.

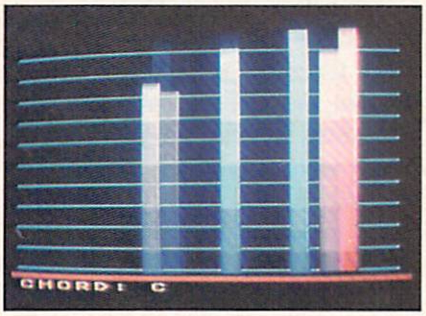
Thus, it is Manhattan's famed skyline that is the target of munching alien mutants in *Save New York.* You, as the intrepid hero, must take to the air in your fighter to clear the sky of monstrous vermin. In addition to the high-flying horrors, you must deal with their metal-eating offspring, the mutant eggs that land in the city's subway and sewer systems. Upon hatching, these junior mutants assist their parents

by gnawing on skyscrapers from the ground up. To clear the little buggers out, you must land your jet and work your way through the labyrinth of underground passageways.

The pace of *Save New York* is frantic, the action constant, and the gameplay enjoyable. This is especially true when you play with another person, and have to watch out not only for the aliens, but for your opponent as well. Another factor that contributes heavily to the frenzy of the action is your fighter's inability to hover. It will always continue to travel in the direction pointed, causing the player whose attention drifts to park from time to time in someone's living room. Also, when blasting the aliens, your aim must be true: your air-to-air missiles will do just as much damage to New York as the aliens.

Save New York? Some folks wouldn't do it on a bet, but it's a sure bet that anyone who misses this game will be missing out on a good deal of fun.

—R.J. Michaels



DANCING FEATS

Softsync, Inc.
C-64
Disk, cassette; joystick

Have you ever played a few notes on one of those electric organs that can back you up with anything from simple percussion to the string, wood, and brass sections of the London Philharmonic? If so, you know that regardless of musical training, with enough electronic assistance it's hard to sound bad.

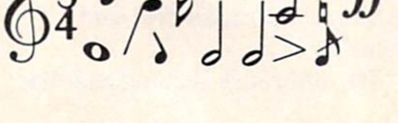
It's even harder with *Dancing Feats.* While you wail away on your joystick to program-provided accompaniment, you'll be playing only notes within certain scales determined by your selection of backup elements. When the chord changes, so do the notes. You can drop to a lower octave by pressing the fire button.

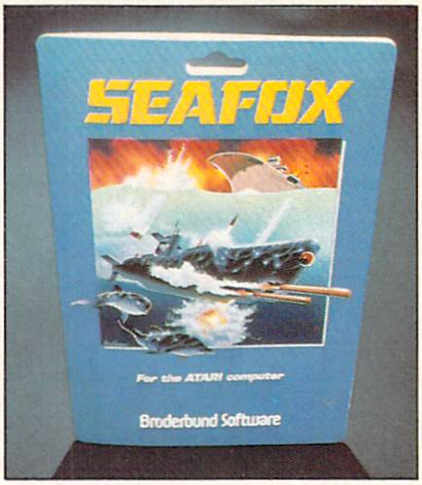
The program allows you to pick your own bass, beat, style, tempo, and ending, with four choices for each. (Ending options, for example, consist of The Duke, The Elvis, The Chance, and The Mozart.)

Each time you play a note, a multicolored bar graph climbs quickly to the top of the onscreen staff. This may tempt you to twirl the joystick as quickly as possible, putting the most color onscreen and generating the most notes. That's enjoyable for awhile, just as listening to KISS is enjoyable for a minute or so. But to produce authentic-sounding music, it's best to establish a simple three-or four-note pattern that's easy to repeat, and change it slightly as you go along, staying in sync with your accompaniment.

The name of each note you play flashes on the screen, as does the name of the chord you're playing in. This feature, among others, makes *Dancing Feats* something of a painless music teacher. After all, the program is as talented in many ways as a professional musician. (We can hear it now: a slew of new hits by Lionel Richie and the Commodores.)

—Steve Springer





SEAFOX

Broderbund Software
C-64 and VIC-20
Cartridge; keyboard, joystick

Though kissing off the claustrophobic market, Broderbund' has produced in *Seafox* a game that fans of WWII subhunt films will set their sights on. Your objective is to destroy the enemy convoy traveling along the surface while avoiding killer subs.

As the game begins, it seems easy enough: torpedo the subs and launch missiles at the cruisers. But that's only Mission One. As you advance to successive rounds, the challenge mounts. In addition to enemy subs that dog your steps, you must dodge magnetic mines and depth charges from PT boats. True to their name, the depth charges will not explode randomly, but at the depth where your sub is located.

Graphics and gameplay are not very elaborate (particularly in the VIC version, whose screen is so stark that it barely held my interest.) Yet *Seafox* should not be dismissed as just another submarine hunt-and-destroy mission. It's always refreshing to find a battle game that incorporates a certain amount of playfulness. *Seafox* does this by making you refuel and resupply with the aid of a

trained dolphin. Also, a hungry clam will swim up from the ocean floor to snack on your supplies.

While the action isn't very fast paced, it warrants continued playing. My main complaint with the gameplay is that you cannot turn your ship around: you will always face the right, forcing you to maneuver underneath U-Boats that approach from the left to aim your missiles at them. Still, this constraint adds to the strategy involved.

—R.J. Michaels

DATABASE MANAGER
Mirage Concepts
C-64
Disk

Joan Rivers turns to her database before going onstage—it reminds her of exactly which jokes she told the last time she played a particular club, enabling her to avoid repeating the same material. The information you store in your database may not lead to as many laughs as hers, but it will be much more accessible and easier to manipulate if you select a program that has the "right stuff." *Database Manager* from Mirage Concepts ($99.95) is packed with more than its share.

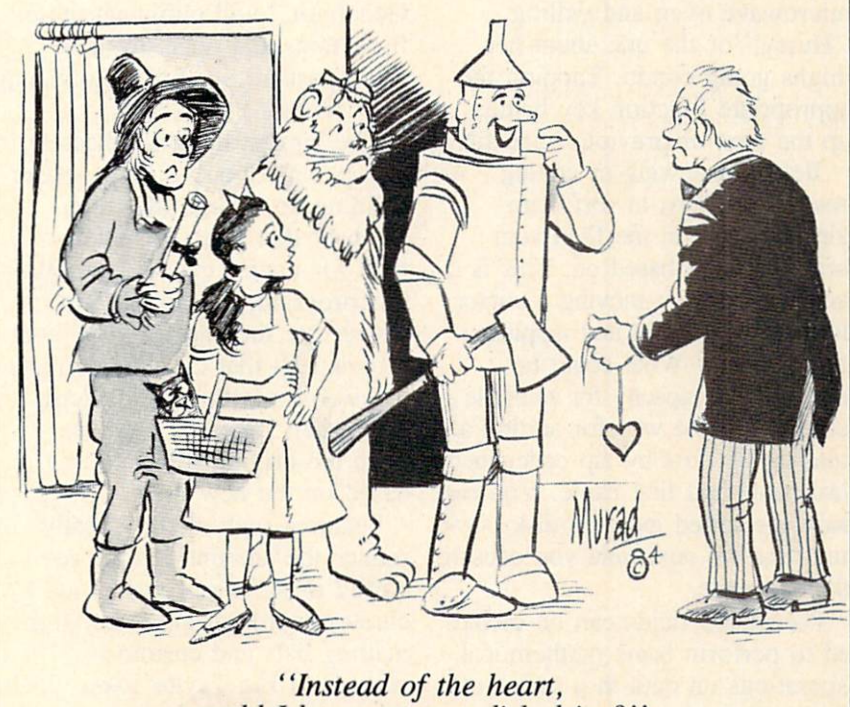
For one thing, it's an ISAM-based program. This type does its work much more efficiently than sequential filing systems, whose files must be loaded into the computer's RAM before they can be consulted or updated. With an ISAM set-up, only a sequential index of the files goes into the RAM. This means you can create larger files (150,000 characters on a single disk), because there's more room on the disk than the 64K of your computer's RAM. Searching for a specific record or set of records is accomplished faster, too—sequential files take longer because the search proceeds from the beginning of the file to the end, or until the object of the search is located.

Of course, sequential databases have one plus—the data can usually be transferred into your word processor, which also reads and writes sequential files. But you get that capability with *Database Manager,* too, because it handily converts any number of records you choose into a sequential file.

If you have little or even no experience with a database, this makes an excellent first package. The manual provides some of the most concise and comprehensible documentation you'll find anywhere. A thorough, step-by-step tutorial and a glossary are included for beginners; the reference section enables seasoned database users to master this system's ins and outs practically overnight.

But just because it's easy to use, don't think this program is oversimplified or lacks flexible and powerful features. The maximum size of a single record is a whopping 2,000 characters, and each record can hold up to 200 fields. Creating the forms that your records are based on is painless. You don't have to specify the number of characters per field; just press the "f1" key and watch a line move across the screen until it's the proper length.

In case you're the forgetful type, the name of the file currently in use is displayed at the top right. A menu remains perched at the top of the screen to remind you exactly what each of the function keys does in that particular mode. The different modes of operation cover writing files, review/edit, sort, print, creating new form and file com-



"Instead of the heart, could I have my own disk drive?"

mands. (Because the menu's always present, you must scroll up and down to view lengthy records.)

If Joan Rivers were looking for a joke she told at Resorts International in Atlantic City, she would first move the cursor to the field she wanted to search in each record, then hit the "f5" key. That would be the "club" field. Then she could review each record whose club field said Resorts International. A far more powerful and useful type of search could be done with *Database*—the conditional search.

This involves searching for a record containing matches in two or more fields. Joan could type in "Resorts International" for the club field and "Liz Taylor" for the "type of joke" field. *Database* would quickly find the first such record. Then she could read the rest of it to find out exactly which joke she told the last time she played there—the one about Liz standing in front of the microwave oven and yelling, "Hurry!"or the one about her thighs going condo. Tapping the appropriate function key brings up the next or previous such file.

Before reviewing or editing records, it helps to sort them alphabetically in the field your search will be based on. This is accomplished by moving a cursor to the chosen field and tapping a function key. What could be easier? The capacity for multiple sorts opens the way for sorting a mailing list first by zip code, then last name and first name. You can save the sorted index to disk, saving time the next time you consult the same file.

Calculated fields can be included to perform basic mathemtical operations on data in a series of fields. You choose a formula, which might be the equivalent of "add the number in field one to that entered in field two, then put the total in field three." Decimal points can be rounded off according to your own specifications.

When it's time to churn out hard copies of your data, several alternatives exist. The first prints a complete copy of each record, or just the ones you select. You may prefer a list that contains only information from specific fields. Finally, a report generator allows you to title the report at the top of each page, then prints your selected fields in columns with a name of your choice at the top of each.

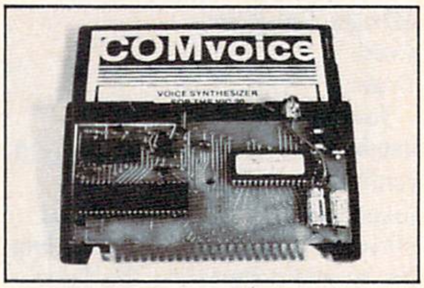
The only drawback is the inability to send control characters to the printer—these could otherwise be used to underline titles and take advantage of your printer's other capabilities. Mailing labels may be printed from information in a larger record, but you're limited to standard 3½ X 15/16″ labels. (For $59.00, an "Advanced Report Generator" will eliminate these limitations and open up many more possibilities for customizing your reports.)

The biggest database disaster strikes when you've spent weeks building up a huge file, then discover that you need another field for unanticipated data. With this program, it's possible to restructure the entire file without retyping all that data. You create a new form with an additional field, then "merge" the data from the original file into one based on the new form.

Features such as this greatly reduce the amount of time required to write and update accurate records of any type, from mailing lists and customer records to Liz Taylor jokes. Joan Rivers may not be using *Database Manager* to file her gags, but it's a very cooperative filing system that's friendly enough for a first-time user yet powerful enough for the office.

—Shay Addams





COMVOICE

Genesis Computer Corporation
C-64, VIC-20

The COMvoice speech synthesizer is based on the Votrax SC-01 chip which uses phoneme synthesis to assemble word sounds from individual speech components. This gives COMvoice an unlimited vocabulary. The product is available in two versions for the VIC-20 and the Commodore 64. The two-part package consists of a speech module, which plugs into the computer's expansion slot, and a separate loudspeaker. The jack for the loudspeaker can be used to drive an external amplifier. The volume using the supplied loudspeaker is adequate for most home applications.

The speech module contains the Votrax chip, a 6522 VIA chip to interface with the computer's expansion bus and an 8 kilobyte ROM chip. It is the program stored in the ROM which makes

the COMvoice extremely simple to use. When the computer is turned on with the COMvoice in place, the user is greeted by a brief "COMvoice ready" message in the distinctive overtones of the Votrax chip. After a very brief delay, the usual Commodore "bytes free" message is displayed. In the case of the VIC-20, nothing else is apparent since the COMvoice Program lives in the $A000 block. This means that all of BASIC RAM is still available and no expansion RAM is needed.

On the Commodore 64 the voice program resides in the $8000 block at the top of BASIC displacing 8 kilobytes of BASIC RAM. Thus the initial bytes free are reduced to 30719. In either case BASIC has been enhanced by the addtion of a SPEAK command. Syntax is very similar to the PRINT command in that any spoken text is simply placed in quotes following the SPEAK command. Alternatively, the text may be transmitted as a string variable. For example:

```
10 INPUT A$
20 SPEAK A$
30 GOTO 10
```

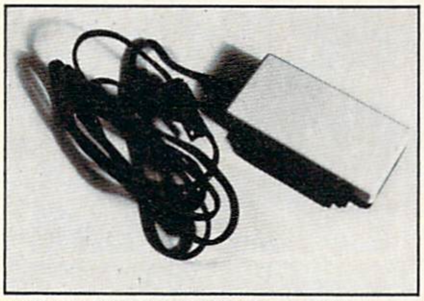
will translate any text typed at the keyboard into speech.

The conversion algorithm performed rather well, correctly translating over 90% of the common and not so common words we tried. The speech is fairly intelligible within the capabilities of the Votrax chip. If anything, the device seems to have a bit of a New England accent. Numbers are spoken as well, provided that they are handled as a string and that they fall in the range of -999,999,999 to +999,999,999.

The 20-page manual is easy to follow. Instructions are included for programming phonemes also.

The COMVoice is easily programmable directly from BASIC. It would be useful in providing voice interaction in educational programs. Price is $149.95. Available from Genesis computer corporation, 1444 Linden Street, Bethlehem, PA 18018.

—Morton A. Kevelson



THE VICONTROLLER

Genesis Computer Corporation
C-64, VIC-20

News flash from the future: users of the BSR remote control switches can now employ their VIC-20 or Commodore 64 as a full featured control center, utilizing those devices.

The BSR remote control switches, which are marketed by Radio Shack under their own brand and by Sears, are available in several configurations for the control and dimming of lights and the switching of appliances. When used with the BSR remote controllers, up to sixteen different operations can be controlled. The actual number of unique remotes which can be addressed by the system is 256; however, this cannot be conveniently done with the BSR controllers. Each remote unit is called by a device code (1 through 16) and by a house code (A through P). Accessing more than one house code with the BSR controllers is inconvenient or would require separate remote controllers set to different house codes. The software supplied with the VIController allows easy access to all 265 codes directly from the computer keyboard.

The VIController is a small plastic package which plugs into the user port in the rear left corner of the computer, making the same product compatible with both the VIC-20 and the Commodore 64. The 120 volt line cord is used only to connect the controller to the house wiring which acts as a communications channel for the high frequency signals (40 kilohertz) used by the BSR system. Power needed for the device is minimal and is supplied entirely by the computer. This power line carrier system is actually very similar to the system used by electric utilities for long distance communication over power transmission lines.

Three software packages are supplied with the VIController. The Manual Control Program allows direct control of all remotes from the computer keyboard. All the functions of the BSR system are supported (on, off, dimming, all off, all lights on). The dimming feature is implemented in ten discreet steps rather than the continuous control provided by BSR. A Time Control Program is provided which allows up to 30 commands to be stored in an unexpanded VIC-20, considerably more with memory expansion or a Commodore 64. This compares favorably with the thirty-two commands available with the BSR timer.

When the abiity to control 265 unique locations (the BSR only addresses eight) and the ability to use the dimming feature (BSR only does on and off) is considered, the power of the package becomes apparent. Add to this the ability to store different schedules on tape or disk and you have a powerful package indeed.

The Time Control Program only works on a 24 hour cycle. For Commodore 64 users an Extended Time Control Program is supplied as well. This program includes a full seven day schedule as well as user friendly prompts and input error checking. This program will be available for the VIC-20 with expansion memory by the time you read this.

A well-written manual is included with complete instructions, some reasonably detailed explanations on how the system works, and some short subroutines with instructions for writing your own customized control programs. The VIController should be especially attractive as a dedicated application for VIC-20 owners who have upgraded to a Commodore 64.

The company also has available a COMsense module (not tested by BMR) which plugs into the joystick port and provides four open-close and two analog-to-digital inputs. This module, when used in conjunction with the VIController, will allow application to residential burglar protection as well as commercial applications.

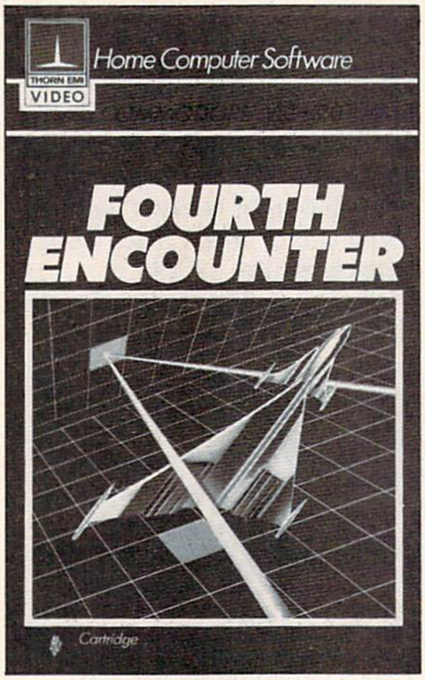
VIController, $69.95. COMsense, $49.95. Genesis Computer Corp., 1444 Linden Street, Bethlehem, PA 18018.

—Morton A. Kevelson

FOURTH ENCOUNTER

Thorn EMI Video Limited
VIC-20
Cartridge

Another *Galaxian*-type attack game, *Fourth Encounter* offers some useful playing features. To begin with, four different types of alien attack patterns with three difficulty levels are presented. The first three are somewhat similar, with the defender restricted to horizontal movement along the base of the screen. The fourth pattern is different in that the attacking aliens are restricted to the screen borders (horizontally and vertically) while the defending ship can roam about the central display area. There is one catch: you can only fire up or down, not left or right. The aliens, of course, are under no such handicaps.



The game has several useful features including one-or two-player option with individual scores and high score displayed. For those who like to try out each game option or need to practice at some specific level, the option to select both the attack wave and the difficulty level is available. Full scoring is maintained for all game selections (a nice feature).

The graphics are limited to the design of the various creatures, but a commendable job was done on that. The third wave was particularly striking at first, as the attacking ships were identical to the defending craft. The arcade action is very good, with plenty of speed and suitable sound effects.

—Morton A. Kevelson

MUTANT HERD

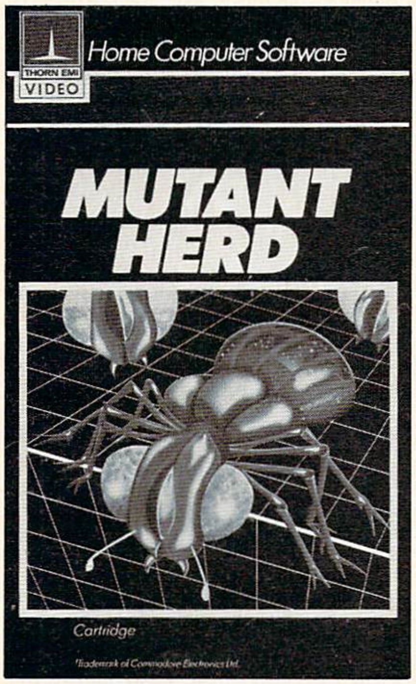
Thorn EMI Video Limited
VIC-20
Cartridge

Here is a two-part arcade-type game with a unique twist. In part one the objective is to keep a swarm of plutonium-crazed mutants away from the Powerhouse in the center of the screen while directing a purple mutant-slayer into one of the entrances to the undergound burrows. To accomplish this, a pair of laser barriers, which look like crosshairs on the screen, are moved by the joystick or keyboard. The player must decide between the two game objectives: protect the Powerhouse or guide the mutant slayers. If too many mutants reach the Powerhouse, or if all your mutant slayers are swept uselessly off the screen, the game ends.

Part two starts when a mutant slayer is guided into one of the burrows. The slayer, which has turned green at this point, must be guided down the ladder to plant a bomb by the mutant queen's eggs. The local mutants make this rather difficult by their annoying tendency to drop rocks on the poor slayer's head.

Once the bomb is planted, the slayer must return to the top level of the burrow where the detonator is located. Success results in the destruction of five eggs and the closing of one of the burrows entrances, at which point you are returned to part one of the game. Difficulty then increases as there are only three entrances open and the laser barriers have developed gaps in them. The whole procedure must be repeated four times to completely destroy the undergound burrows.

Mutant Herd makes extensive use of the VIC-20 graphics, sound and color capabilities. The crazed mutants resemble a swarm of skit-



tering insects and the incessant chittering sounds tend to make one's skin itch. Depending on your mood, this can be a definite plus. Animation and game action worked rather well. The game will support one or two players and retains the high score.

—Morton A. Kevelson

SIMONS' BASIC
Commodore Software
C-64

If you do any programming in BASIC and should happen to see this product on a dealer's shelf, do not ask any questions—do not hesitate—just buy it! *Simons' BASIC* is probably one of the most powerful extensions to BASIC available for any personal computer presently on the market. Its price makes it one of the biggest bargains available for the Commodore 64.

This 16-kilobyte cartridge extends the C-64 BASIC with 114 commands. These commands are grouped into ten categories by the 151-page manual. All this is accomplished at the expense of 8 kilobytes of BASIC RAM by bank switching the cartridge. When the Commodore 64 is powered up with the cartridge in place, the bytes free message is reduced to 30,719 bytes.

For BASIC programmers there is a full set of programming aids including function key assignment, automatic line numbering and line renumbering, search for BASIC code or strings, trace, dump, error trapping and program merge from cassette or disk. There are program listing aids which include screen paging and variable speed screen listing. There are even two security commands, DISAPA and SECURE, which prevent listing of selected program lines. The renumber feature does not include renumbering of GOSUBs and GOTOs, something of a nuisance with existing programs. This is not as serious as you might think, as *Simons' BASIC* supports structured programming.

Structured programming allows control of program execution without the use of line numbers. These techniques, which are used by other high-level languages such as Pascal and Logo, allow the calling of subroutines and procedures by name rather than by line number. Included as well are condition testing and program loops such as IF ...THEN...ELSE, REPEAT...UNTIL and LOOP...EXIT IF...END LOOP. There is even the capacity to define local and global variables. For more conventional BASIC programmers, the capability for computed GOTO is available in *Simons' BASIC.*

Text manipulation has been greatly enhanced with commands to merge and overwrite strings, determine position on substrings within strings and duplication of strings. The CENTRE command betrays the British origin of *Simons' BASIC* as well as centering text on a screen line. The A1 command allows printing by screen position, eliminating all those endless strings of cursor movement controls. There is even a form of print using which has always been missing from Commodore BASIC. Input validation commands have been added to allow easy checking of inputted text for proper content.

Math functions have been extended with the addition of commands which return the results of a division operation as either an integer and the remainder as an integer or as a fraction (MOD, DIV, and FRAC). Binary to decimal and hexadecimal to decimal conversion are available as well.

Disk operations have been enhanced with a DISK command which opens the command channel and transmits a command string to the disk drive. The DIR command provides a directory listing to the screen without destroying the program in memory. Users of the disk wedge will be familiar with these techniques. While these commands are not as powerful as the wedge commands, they offer sufficient improvement over BASIC 2.0 to make the wedge almost unnecessary.

Simons' BASIC includes a variety of graphics and screen manipulation commands. For graphics there is high resolution and low resolution plotting with extension commands such as COLOUR (that British origin again), PLOT, CIRCLE, ARC, PAINT, DRAW and ROT. The CHAR and TEXT commands allow printing text on graphics screens. The REC command plots a rectangle by specifying the coordinates of one corner and its length and its width. The BLOCK command does the same for a filled-in rectangle. Screen manipulation includes background and border color selection,

FLASH which flashes selected characters into reverse and back and BFLASH which flashes the border. There are commands to fill a section of the screen with a specified character and a specified color or both and a command to duplicate specified sections of the screen at another screen location. There is even a command to save and load low resolution or character screens as sequential files to tape or disk. Topping things off in this category are the much sought after screen dumps to the VIC graphics printer including both a high resolution dump (COPY) and a test dump (HRDCPY).

Sprites and user-defined character sets can be programmed directly with *Simons' BASIC.* A set of commands allow the allocation of the required memory, movement of the existing character set and definition of the sprite or custom character by simply typing its image in a set of consecutive program lines. Gone are the hours of tedious calculations or the need for complex character editor or sprite generator programs. The DETECT command is used to sense MOB collisions and the MMOB command displays and moves sprites. The game controller ports are supported as well with commands to read the light pen, paddles and joysticks.

Sound on the Commodore 64 is controlled by five commands in *Simons' BASIC.* VOL sets overall loudness. WAVE sets waveform, controls synchronization and ring modulation. ENVELOPE sets the attack, decay, sustain and release times. MUSIC allows writing songs using the letter keys to define notes and the function keys to set note duration. PLAY allows playing of music while the program continues to execute or while pausing the program till the end of the song.

One of the best things about *Simons' BASIC* is the excellent manual. It is very well organized with numerous examples to illustrate the many commands. Even more amazing is the fact that every example we tried out actually worked. The manual is not a tutorial on programming with *Simons' BASIC* in that a knowledge of BASIC programming is assumed. However, the detailed explanations of the way graphics and sound are handled by the Commodore 64 add to the understanding of how the commands are used. A separate chapter with program examples is included.

Simons' BASIC will be available from Commodore for $29.95.

—Morton A. Kevelson

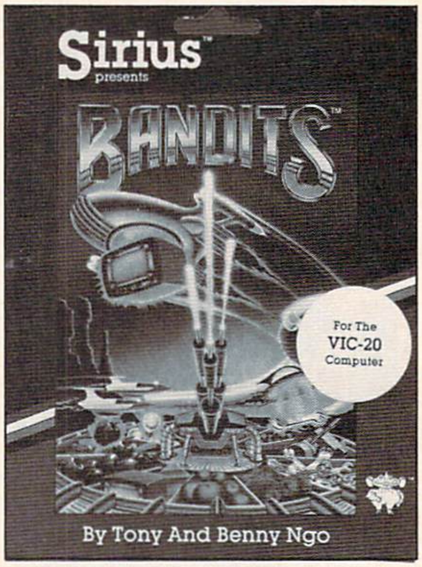
(Note: for a complete listing of the 114 commands provided by *Simons' BASIC,* turn to page 94 in our program listings section.)

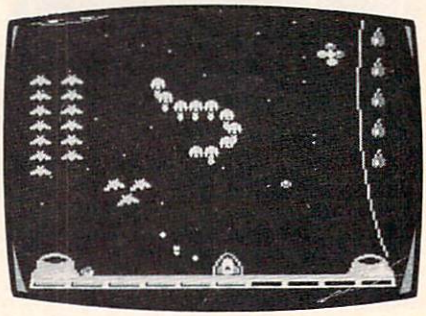
BANDITS

Sirius Software
VIC-20
Cartridge
(Also available on disk for the Commodore 64

While bearing some resemblance to *Galaxian,* Bandits offers a good combination of graphics, color and sound with some fast action on the VIC-20. Your job is to guard your supply base from a variety of criminal invaders who are out to get you and your supplies.

The aliens attack in fast moving waves swooping erratically about the screen. Your single laser cannon is restricted to horizontal movement along the bottom. Protection against all attacks is available in the form of a force shield activated by the joystick; however, a limited supply of energy is available for it. Once the supply runs out, the shield will not be available until your next ship.

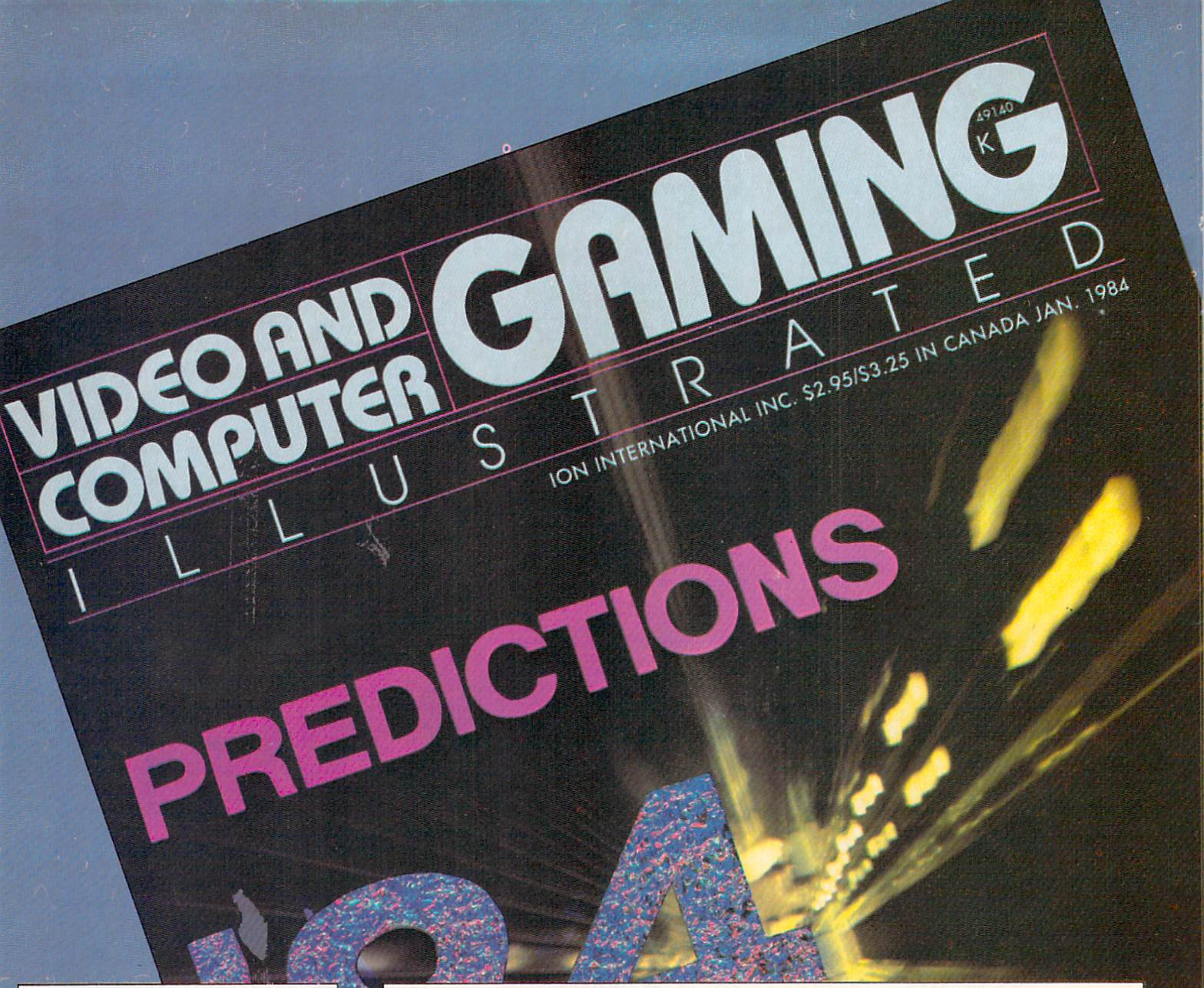




You will by plagued by a host of aliens including three types of thieves, a group of Carriers that will bombard you with bouncing Nuisants, and an occasional Menace to liven things up. Scoring is based on hits, with extra points if you nail a Bandit stealing your supplies. Bonus points are awarded for remaining supplies after each wave. If you lose all your supplies, the game is over. Along with the difficulty increase that occurs with each wave comes a corresponding change in your supplies, adding an interesting wrinkle.

Overall, the game action was good with appealing graphics and appropriate sound. a horizontally scrolling starry background adds to the effect. The VIC-20 version did not retain the high score.

—Morton A. Kevelson



NEW PRICE FOR 1984— $19.84

Sure, the cost of living has scrolled up right off the screen! But that needn't stop you from enjoying all *Video and Computer Gaming Illustrated* has to offer! You can receive twelve bi-monthly issues—enough to carry you well into 1986—for just $19.84! If that's too much of a bargain for you, you can receive one year—six bi-monthly issues—for $12.00!

Either way, you'll be sure not to miss an issue of *Video and Computer Gaming Illustrated*—featuring late-breaking industry news, shrewd game strategies, insightful interviews, exhaustive game reviews, and some of the finest horn-blowing anywhere!

No one playing with a loaded disk drive would pass up an offer like this! Enter my subscription to *Video and Computer Gaming Illustrated* for (check one):

- ☐ Six issues for $12.00 ($18.00 Canada and elsewhere)
- ☐ Twelve issues for $19.84 ($23.84 Canada and elsewhere)

Name ______________________________

Address ______________________________

City ____________ State ____________ Zip ____________

Send coupon or facsimile to:
Video and Computer Gaming Illustrated
45 West 34th Street—Suite 407
New York, NY 10001

TO OUR READERS

SHOOT US!

Nobody's perfect, the conventional wisdom goes—and far be it from us to buck conventional wisdom. Still, we never intended to validate that old saying as irrefutably as we did in our first issue.

Did we get letters and calls—by the hundreds, from all over the United States and Canada. Readers were incensed—and rightly so. After spending hours (days?) typing in the programs featured in the premier issue of *AHOY!*, they found that one of them *did not run!*

We hasten to point out that Michael Kleinert and David Barron are in no way to blame for the failure of their sequential files program to perform. That was an in-house botch, for which we apologize to our readership. Here is the correct way to type in the two bogus lines:

```
360 GETA$:A=VAL(A$):IFA<1ORA>5THE
N360
450 PRINTCHR$(147);"ENTER NAME, T
O DELETE:":N$="":INPUTN$:IFN$=""T
HEN230
```

Pete Lobl, who communicates with us only during the fifteen minutes or so each month when all his mainframes are down simultaneously, alerted us several days after we went to press of a bug in his Interrupt Music Maker/Editor that makes it impossible to leave the program without crashing. He tells you all about it in this issue's installment of the Music Maker program.

Concerning Pete's already classic Multi Draw 64 program (a.k.a. the 64 Graphics System), several readers pointed out that there is no statement in BASIC that reads:

```
IFKTHEN
```

They're almost right. That is merely an abbreviation of the statement:

```
IFK<>0THEN
```

Various other readers could not figure out how to enter the joystick pattern listed in lines 874-878 of Multi Draw 64. Experienced hackers knew to do it like so (shifting for the underlined characters):

```
874 PRINT"   M B N"
876 PRINT"  JCC+CCK    KEYS TO MOV
E CURSOR"
878 PRINT"   N B M     ( OR USE JO
YSTICK )"
```

Additionally, the failure of our daisy wheel printer to reproduce two graphics characters robbed Multi Draw of a fraction of its incredible capability. We're not blaming our printer—the error should not have escaped our notice. (If it's any consolation to the victims of these two in-house bollixes, the appropriate head has rolled.) Here's how to enter the lines in question:

```
147 IFA$="↑"THENA=PEEK(689):A=(A+
1)*-(A<15):POKE689,A:GOTO100
895 PRINT"£ = TOGGLE FAST/SLOW CU
RSOR MOVEMENT"
897 PRINT"↑,* = INC/DEC TEXT BACK
GROUND"
```

Nor were we above making mistakes in the section of our magazine written in English. Not only did we spell Protecto Enterprizes with an 's' instead of their famous 'z' in our interview with Bill Badger—we also misspelled Bill's name and that of Protecto's president, John Scheele!

Once again, we apologize. Pete Lobl apologizes. And we know a headless keypunch operator who we're certain would apologize—if he could.

WANTED: LETTERS FROM READERS

We're interested in what you think—and so are your fellow readers. Write about the new printer interface that has helped you to find inner peace, or why your users group is the baddest on the east

Continued on page 70

Commodore Screen Manipulation

By Michael Kleinert and David Barron

Last time we gave you a general overview of the Commodore 64's memory arrangement. In this article we will show you how to use the PEEK and POKE commands to manipulate screen memory. What are PEEK and POKE, you say? Read on and you'll find out.

Screen memory ranges from memory locations 1024 to 2023. These memory locations keep track of what character is stored in each individual position of the screen. Because the Commodore 64 is capable of displaying characters in different colors, the computer must also store each character's color in a certain place in memory. This area of memory ranges from 55296 to 56295. Both memory locations 1024 and 55296 refer to the same character on the screen. This happens to be the first character which is located in the upper left hand corner of the screen. As you may have already figured out, locations 2023 and 56295 both refer to the last character displayed on the screen, which is in the lower right hand corner. The memory locations in between control consecutive screen locations. In other words, the second character on the screen is kept track of by locations 1025 and 55297. This continues all the way across the first line of the screen. Once the last character of the first line is reached, the next memory location refers to the first character of the second line of the screen. This is better illustrated by the diagram at the top of the following page.

The unparenthesized number in each case indicates the screen character locations, and the parenthesized number the corresponding screen color locations.

At this point, you are probably wondering how you use PEEK, POKE, and all of these crazy numbers to manipulate characters on the screen. Before we talk about manipulating characters on the screen using POKE, let's talk about the command itself for awhile.

POKE allows you to place a number from 0 to 255 into a memory location ranging from 0 to 65535. The format for this command is:

```
POKE (MEMORY LOCATION),(DATA)
```

For example, if we wanted to put the number 200 into memory location 5000, we would type:

```
POKE 5000,200
```

```
LINE 1    1024(55296)                              1063(55335)
LINE 2    1064(55336)                              1103(55375)



LINE 25   1984(56256)                              2023(56295)
```

You should not POKE random data into random memory locations, because you might disturb vital information which the computer uses to keep track of things. Sometimes this may result in a nonrecoverable "crash" and you would have to power down, losing any program you might have in memory.

If a memory address or value is out of the above specified range, the result will be an "?ILLEGAL QUANTITY ERROR".

At this time, we should also discuss the PEEK command. Its format is:

```
PEEK(MEMORY LOCATION)
```

where the memory location ranges from 0 to 65535. However, this is a numeric function and cannot be used alone. It may be printed, assigned to a variable, or operated on mathematically. Its function is to return the value stored in the specified memory location. For example, if we had previously POKEd 200 into memory location 5000 as illustrated above, then the statement PRINT PEEK (5000) would print 200 on the screen. Go ahead and try it, but make sure there is no important program in memory.

Now that you know how to PEEK and POKE, let's apply this knowledge to the screen. Try the following example:

```
POKE 1024,42:POKE 55296,0
```

This will put an asterisk (*) in the upper left hand corner of the screen. It will be black because the 0 that we POKEd into its color memory location (55296) represents the color black. If you happened to do this with a black background, you would not see anything, since the character would blend in with the rest of the screen. Following is a list of all available colors with their corresponding values.

```
BLACK         0
WHITE         1
RED           2
CYAN          3
PURPLE        4
GREEN         5
BLUE          6
YELLOW        7
ORANGE        8
BROWN         9
LIGHT RED     10
GRAY 1        11
GRAY 2        12
LIGHT GREEN   13
LIGHT BLUE    14
GRAY 3        15
```

If we wanted to put a light green asterisk in the upper left hand corner, we would have said:

```
POKE 1024,42:POKE55296,13
```

You are probably wondering how we know that a 42 is an asterisk. We looked this up in the *Commodore 64 User's Guide* which comes with your computer. There is a convenient chart on page 132. The same chart can be found on page 376 of the *Programmer's Reference Guide*. Do not confuse screen display codes for PEEK and POKE with ASCII and CHR$ codes. They are not the same, and have different purposes.

To find the memory location of a specific character on the screen, use the following formula:

```
MEM = 1024 + COL + (ROW * 40)
```

where COL is the column and ROW is the row. For the corresponding color memory location, just use the same formula, replacing the 1024 with 55296.

(NOTE: The first column and row on the screen are referred to as row and column 0, not 1.)

Let's write a short program to demonstrate what we have learned so far. Our program will fill up the entire screen with asterisks, and then erase them all, using the POKE instruction.

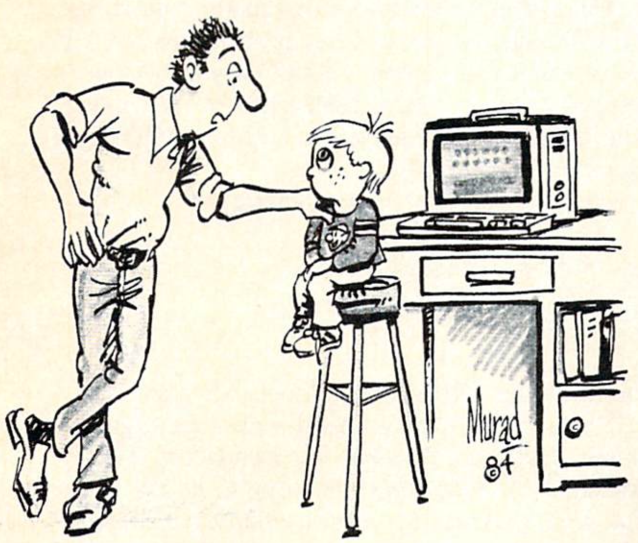
```
10 FOR X = 0 TO 1023 : REM 1024 L
OCATIONS ON SCREEN
20 POKE 1024 + X,42   : REM PUT AS
TERISK IN CHARACTER LOCATION
30 POKE 55296 + X,0   : REM MAKE T
HE ASTERISK BLACK
40 NEXT X             : REM DO THE
 NEXT LOCATION
50 :REM SCREEN IS NOW FILLED
60 FOR X = 0 TO 1023 : REM 1024 L
OCATIONS TO ERASE
70 POKE 1024 + X,32   : REM PUT A
SPACE WHERE THE ASTERISK WAS
80 NEXT X             : REM ERASE
NEXT LOCATION
90 END
```

Notice that the second FOR...NEXT loop does not contain a POKE to a screen color location. This is because a space does not show up on the screen, and does not require a color.

Sometimes it is convenient to first color the entire screen before beginning to POKE characters onto it. The following routine will do just that. You can put it at the end of your programs and call it with a GOSUB statement. (NOTE: If the screen is cleared, it must be recolored.)

```
9000 FOR X = 55296 TO 56295 : REM
 RANGE OF COLOR MEMORY
9010 POKE X,C               : REM
 SET C EQUAL TO DESIRED COLOR
9020 NEXT X                 : REM
 REPEAT UNTIL DONE
9030 RETURN                 : REM
 GO BACK TO MAIN PROGRAM
```

That should be enough to keep you busy for awhile! Next time we will discuss further applications of the PEEK and POKE commands.



"Son, it's time you and I had a little man-to-man interface."

MAN ON A STRING

By Dale Rupert

Computers only work with numbers. They really don't understand things like letters and words, let alone sentences and paragraphs.

Hogwash! This computer I'm typing on is able to perform some spectacular alphabetic feats. It seems to be able to find instantly every occurrence of the word "word" in a 2000-word article. It seems to be able to take a list of inventory items and sort them by name, location, owner, or any other conceivable classification. It seems able to go through a massive text and identify most misspelled words.

This month we'll look at some of the things the computer does with letters and words. To be more specific, we'll deal with strings and string functions We'll start out with some basic string concepts in BASIC, and end up with another look at machine language programming.

First of all, a string variable in BASIC is recognizable by the dollar sign following its name. Examples are A$ and P3$. In fact the dollar sign is usually read "string" as in "A string" and "P3 string." The $ tells the computer and the programmer that the information represented by the variable is not numeric.

It is easy to prove that strings are treated differently from numbers in the computer. Type this program and run it:

```
10 A$ = "7"
20 A  =  8
30 PRINT A$ + A$
40 PRINT A + A
```

Better yet, predict the results. Line 30 is an example of a two-dollar concept with a thirty-dollar name: concatenation. Look it up. It's a real word. The computer doesn't really add strings together in a mathematical sense. It merely concatenates them.

Notice a couple of things about that simple program. First, A$ and A are totally different as far as the computer is concerned. Second, the "$" operator has two distinctly different functions.

Now we know how to build large strings from smaller ones. The 7 in line 10 of the program above could be replaced by anything we could type on the keyboard. The same is not true for the 8 in line 20. Try it and see what happens.

Rather than spend more time building up strings, we will work on ways of taking them apart. Once we've taken them apart, we can do whatever we want and then put them back together. We'll start with the most powerful string function, MID$ (pronounced 'mid-string').

Type and run this short program:

```
10 A$ = "ABCDEF"
20 B$ = MID$(A$,2,3)
30 PRINT B$
```

Evidently the MID$ function took A$ apart. The MID$ function is usually used with three arguments although only the first two are mandatory. The first argument (A$) identifies the string to be taken apart. The next argument (2) tells the starting position within A$. The third argument (3) tells how many characters starting at position 2 are to be taken. Line 20 assigns the string 'BCD' to the variable B$.

If the third argument is missing such as in MID$(A$,2), you get the entire right-hand portion of A$ beginning with the second character. This is not to be confused with the RIGHT$ function. Do you

know the difference?

Change line 20 above to

```
20 B$ =RIGHT$(A$,2)
```

and run it. The second argument in the RIGHT$ function tells how many characters to take (just as the third argument in MID$ does). Consequently the last two characters of A$ are taken with RIGHT$(A$,2) whereas all characters from the second one on are taken with MID$(A$,2).

To round out our repertoire, there is a LEFT$ function. It behaves just like RIGHT$ except the specified number of characters are taken from the left end of a string instead of from the right.

To make sure you understand these functions, predict what the outcome of this program will be. Then run it to prove yourself correct:

```
10 A$="TESTING"
20 B$="GLOVE"
30 C$="BASICALLY"
40 PRINT MID$(A$,5,1),RIGHT$(B$,4
),LEFT$(C$,5)
```

LEN is a very handy function. All it does is tell the length of the string given as its argument. But that can be very useful. For example LEN("WXYZ") is 4, and if A$="COMMODORE", then LEN(A$) equals 9.

Do you notice something significantly different between LEN and the other string functions we've discussed? If you are a keen observer, you'll notice that the function name is LEN, not LEN$. That should tell you that this function gives a numerical result, and it does.

While we are talking about lengths of strings, there's one important string that has no length. It is the "null string." Its most common application is in conjunction with the GET statement. For example,

```
10 GET A$ : IF A$ = "" THEN 10
20 PRINT A$ : GOTO 10
```

GET causes the computer to read the keyboard buffer. If no key has been pressed, then the buffer is empty. Another way of saying it is that the buffer is full of null characters. The quotation marks side by side in line 10 represent the null character. If that's all the computer sees when it reads the buffer, it loops back to the start of line 10 and looks again. This continues until a key is pressed. A$ will store the character that was pressed, and the IF statement will no longer be true. Note that the PRINT in line 20 is necessary to see what was typed since the GET statement does not output anything to the screen.

You can add a password feature to a program by *not* printing the results of a GET statement. Try this:

```
5 GET J$ : IF J$ <> "" THEN 5 : R
EM EMPTY BUFFER
10 PW$="XY"
20 PRINT "TYPE THE PASSWORD :",
30 GET A$ : IF A$ = "" THEN 30
40 GET B$ : IF B$ = "" THEN 40
50 IF A$ + B$ = PW$ THEN 70 : REM
 PASSWORD OK
60 PRINT "SORRY" : STOP : REM PAS
SWORD NO GOOD
70 PRINT "WELCOME" :REM PROGRAM S
TARTS HERE ...
```

Line 5 is necessary to clear out the buffer. The password is defined in line 10. Lines 30 and 40 put one character each into A$ and B$. Line 50 concatenates them and compares the result to the actual password. You would probably change the STOP in line 60 to a NEW statement, so the program is erased if illicit accesses are made.

Perhaps you have come up with some more sophisticated password schemes. If so, let me know about them. I'll reveal them to the world only with your permission.

Let's take a word apart and put it back on the screen vertically. The LEN statement is frequently used with FOR-NEXT statements as in the following program:

```
10 INPUT "ENTER A WORD" ; W$
20 FOR L = 1 TO LEN(W$)
30 PRINT MID$(W$,L,1)
40 NEXT L
```

The user types a word which is stored in W$ in line 10. Line 20 defines the loop repetitions to equal the number of letters in W$. Line 30 takes W$ apart letter by letter. L represents the position of the next letter in W$ and the 1 tells how many letters to print, starting at position L. Try to predict the outcome if there were a comma or a semicolon at the end of line 30.

To print the word upside down, change line 20 to

```
20 FOR L = LEN(W$) TO 1 STEP -1
```

How about printing the word sdrawkcab—I mean,

backwards. That should be a good problem for you. No doubt there are several ways to do it, just with the functions we've covered.

When you are working with strings, one of the more common errors is in trying to perform numerical operations on strings or vice versa. The computer replies "TYPE MISMATCH." If that happens, see where you have a number (string) where a string (number) should be.

There are many more things to do with strings. For now we'll jump into another example of machine language programming. Once again we'll see how to use one of the kernal routines. We will use string functions to convert hexadecimal numbers into decimal.

We will use the other capability of the PLOT routine discussed last month. That is, we will read the screen position of the cursor. Last month we placed the cursor at a specific location. According to information in the Programmer's Reference Guide, if the Carry Flag is set when PLOT is called, PLOT returns the current cursor position in the X and Y registers. We will transfer the results from the registers into memory locations so that we may PEEK the values. (Registers are inside the microprocessor and may not be directly accessed from BASIC. Memory locations are directly accessible.)

Here's the assembly language program:

```
LOC1              ;MEMORY STORAGE LO
CATION
LOC2              ; DITTO
    SEC           ;SET CARRY FLAG
    JSR PLOT      ;CALL THE ROUTINE
AT      $FFF0
    STX LOC1      ;PUT THE X VALUE I
NTO MEMORY AT LOC1
    STY LOC2      ;PUT THE Y VALUE I
NTO MEMORY AT LOC2
```

Looking up the op-codes in Appendix L or starting on page 235 of the Programmer's Reference Guide gives these results in hexadecimal: SEC = 38, JSR = 20, STX = 8E (absolute), and STY = 8C (absolute). Absolute means we will specify a two-byte memory address for LOC1 and LOC2.

Here's the translation from mnemonics to hexadecimal:

```
LOC1
LOC2
    SEC           38
    JSR PLOT      20 F0 FF
    STX LOC1      8E -- --
    STY LOC2      8C -- --
    RTS           60
```

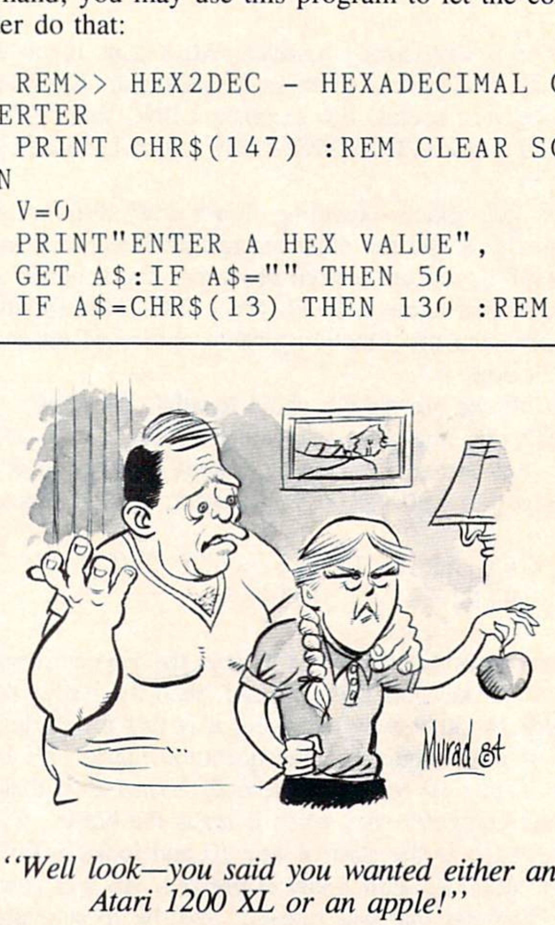
Notice we have a problem similar to forward jumps in a BASIC program. Until you've written the program, you don't know what the line numbers you are jumping to will be. Here we don't know what the addresses of LOC1 and LOC2 are until we decide where we will put this routine into memory. If we start it at location $C010, now we can fill in the blanks. LOC1 is at address $C010 and LOC2 is at address $C011. Here then is the entire program:

```
00  00  38  20  F0  FF  8E  10
 C0  8C  11  C0  60  (HEX)
 0   0  56  32 240 255 142  16
192 140  17 192  96  (DECIMAL)
```

There is no need to put any value for LOC1 and LOC2, but we must leave room for them. Notice that the C010 for LOC1 is stored as 10 C0 and the C011 for LOC2 is stored as 11 C0 as required.

Rather than convert the hex values into decimal by hand, you may use this program to let the computer do that:

```
10 REM>> HEX2DEC - HEXADECIMAL CO
NVERTER
20 PRINT CHR$(147) :REM CLEAR SCR
EEN
30 V=0
40 PRINT"ENTER A HEX VALUE",
50 GET A$:IF A$="" THEN 50
60 IF A$=CHR$(13) THEN 130 :REM R
```



"Well look—you said you wanted either an Atari 1200 XL or an apple!"

```
ETURN
70 IF A$>="0" AND A$<="9" THEN D=
VAL(A$):GOTO 100
80 IF A$>="A" AND A$<="F" THEN D=
ASC(A$)-55:GOTO 100
90 GOTO 50 :REM IGNORE BAD CHARAC
TER
100 V=16*V + D
110 PRINT A$;
120 GOTO 50
130 PRINT " = ";V : PRINT
140 GOTO 30
```

Note that there is no space between the quotation marks in line 50 (the null string)! Run this program and type a hex number of any length, then press RETURN to see the decimal equivalent. Typing C010 gives the starting address of our program as 49168. Entering the other hex values gives the decimal equivalents shown in the DATA statement in line 120 of Listing 1.

Lines 30 through 120 of Listing 1 put the machine language program into memory and prints the values on the screen. Line 150 calls the subroutine at line 200. The SYS SA+2 statement in line 200 calls the machine language routine which begins at 49168 + 2. Memory locations 49168 and 49169 will store the row and column coordinates of the cursor. Lines 210 and 220 read the values from those locations into ROW and COL. Line 230 brings us back to line 160 where the cursor coordinates are printed.

Line 170 TABs the cursor over to column 20 and calls the subroutine again. Now the COL value is 20 and the ROW is one greater than before. Those values are printed by line 180.

With the speed and power of the kernal routine PLOT at your command now, perhaps you can create the world's greatest word processor and prove that computers really do work with letters and words!

```
5 REM << CRSRLOC - LOCATE CURSOR
>>
10 REM <<LISTING 1 - 9/9/83>>
20 REM USE KERNAL ROUTINE 'PLOT'
TO READ THE CURSOR'S POSITION
30 PRINT CHR$(147) :REM CLEAR SCR
EEN
40 REM STORE MACHINE LANGUAGE IN
MEMORY
50 SA = 49168 :REM STARTING ADDRE
SS
60 X=0
70 READ B
80 IF B=-1 THEN 130
90 POKE SA+X,B
95 PRINT SA+X,B
100 X=X+1
110 GOTO 70
120 DATA 0,0,56,32,240,255,142,16
,192,140,17,192,96,-1
130 REM---------------------------
------
140 REM FIND OUT WHERE CURSOR IS
NOW
150 GOSUB 200
160 PRINT ROW;COL
170 PRINT TAB(20):GOSUB 200
180 PRINT ROW;COL
190 END
200 SYS SA+2: REM GET CURSOR LOCA
TION
210 ROW=PEEK(SA)
220 COL=PEEK(SA+1)
230 RETURN
```

OUR READERS

Continued from page 61

coast, or how that software program our reviewer praised actually stunk to high heaven—but write! If you're still talking to us, that is.

WANTED: VIC AND 64 PROGRAMS

Whether your programming forte be games, utilities, or description-defying entities like Pete Lobl's Music Maker—whether your program runs several dozen lines or several thousand—whether geared to the beginner or the expert—we'd be interested in looking at what you have to offer. Send us a copy of your program on disk or tape, with a printout and a stamped, self-addressed mailer. (Program printouts not accompanied by a disk or tape version of the program will be returned unread; programs not accompanied by sufficient return postage will not be returned.)

AHOY! pays competitive rates for programs.

IMM/E

Continued from page 39

procedure for Voice 1; change ADSR, then AUTO, then enter the music. It will take awhile, but it will be worth it.

To play the music type (*P) for play. Answer the questions that will be asked—waveform, etc. Then just sit back and listen. You can edit while the music is playing (since it's on interrupt) and listen to it on the fly. To turn off the play feature simply type another play command, *P, and the music will stop. To change the speed of the music while it's playing, enter the speed command, *2, then give the new speed. 57 is normal; lower numbers will speed up the music, higher numbers will slow it down. Experiment for the best results.

That's all for this edition. Next time we will go over each command separately so you can use the music editor to its full potential.

SEE PROGRAM LISTING ON PAGE 92

By David Ritchie

THE GOLDEN GATEWAY

In an operating room in the not-too-distant future, a patient is on the table for a brain operation. At first glance nothing looks too unusual about the surgical procedure—until one notices the small plastic object on the surgeon's table. It looks somewhat like a plug-in attachment of the kind used on stereos and home computers, only this one is larger and more complex, and it has no wire prongs to fit into receptacles; instead it has tiny wirelike projections made out of stiff organic compounds. In a few minutes this "plug" will be part of the patient's head.

Slowly and painstakingly, the surgeon puts the plug in place upon the subject's left temple. A flange holds the object firmly against the bones of the skull. Set in place, the little attachment looks rather like the bolt through the neck of Frankenstein's monster, in the movies—only this "bolt" appears to be passing through the frontal lobes of the brain. But the implant is small and will be unobtrusive, once the patient's hair has grown back long enough to cover it.

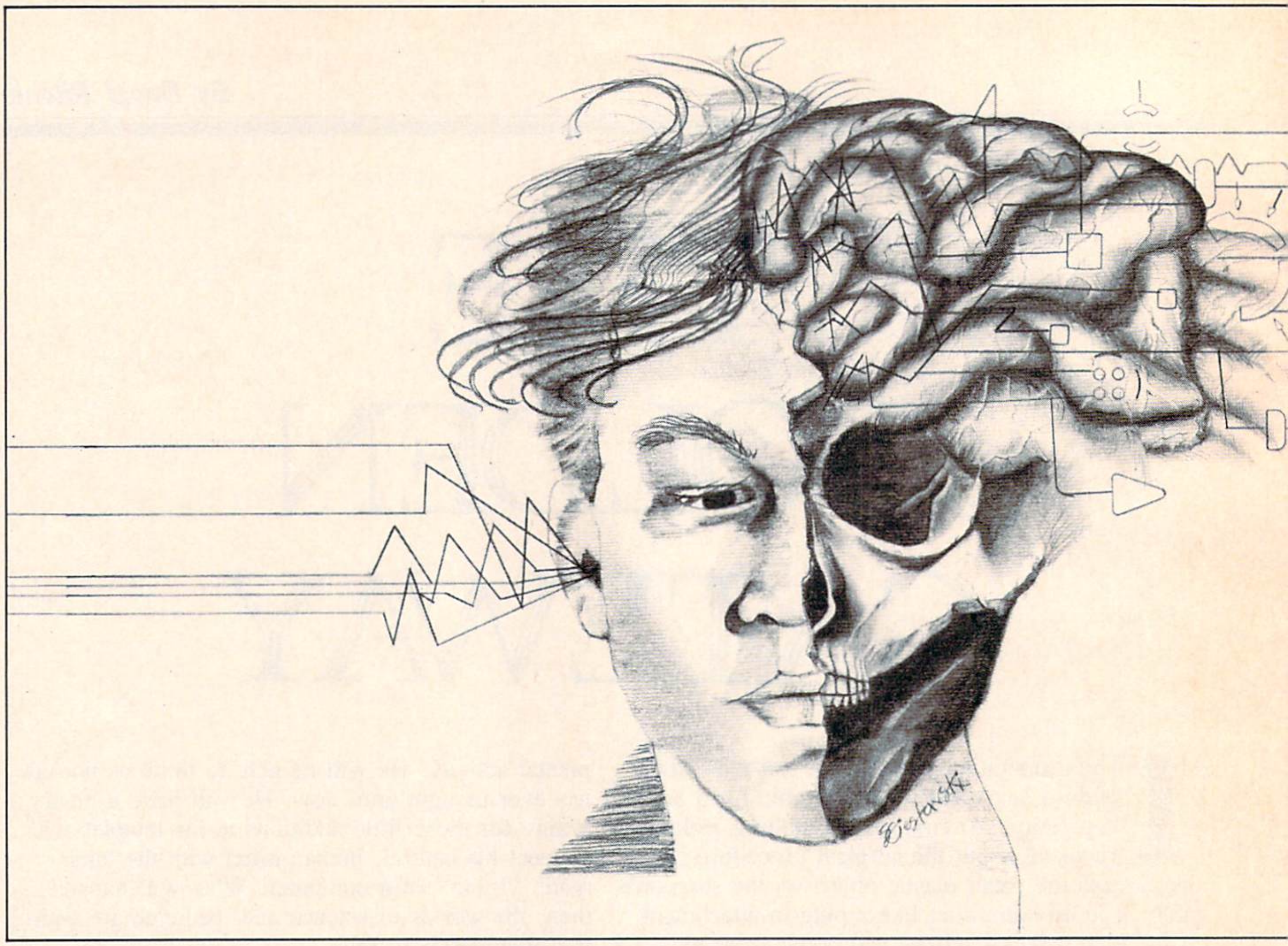
The surgeon finishes installing the first unit in the patient's head, then repeats the procedure on the other temple. The end result is two small flesh-colored protuberances on opposite sides of the skull.

Again, this patient has remained conscious through the operation, his comments helping direct the surgeon's moves. But at the end of this surgery, the patient will be quite a different person than he was before. He will suddenly have access to a vast and heretofore unexplored new world of mental activity. He will be able to think as no one has ever thought until now. He will have a binary brain—for those little additions to his temples will connect his natural, human mind with the man-made "mind" of a computer. What will happen then, the worlds of science and medicine are waiting breathlessly to see.

Scenes very much like this one may take place within our lifetimes—perhaps within the very near future—because of a new field of computer technology, so fantastic and staggering in its implications for our future that it sounds more like wild fantasy than fact. It is known by several technical names, among them *organic data procesor.* Most call it simply a biochip.

A biochip is precisely what its name implies: a microprocessor built along biological lines rather than out of nonorganic materials such as silicon and gallium arsenide. Biochips are nothing new in the natural world; they have existed since the first specialized nerve cells developed, back in the Paleozoic. Humans, however, have improved considerably on the design of natural "biochips," shrinking their dimensions and packing more computing power into them. The man-made biochip operates on the same cybernetic principles as the gray cells in your brain—only the biochip does its job much more quickly. And as we will see in a moment, a practical biochip would have many advantages over silicon-based computers as well.

Although biochips are made up of complex organic chemicals, their structure and function are basically easy to understand. In fact, you can use a



few common items found in the kitchen to build a model of a biochip unit.

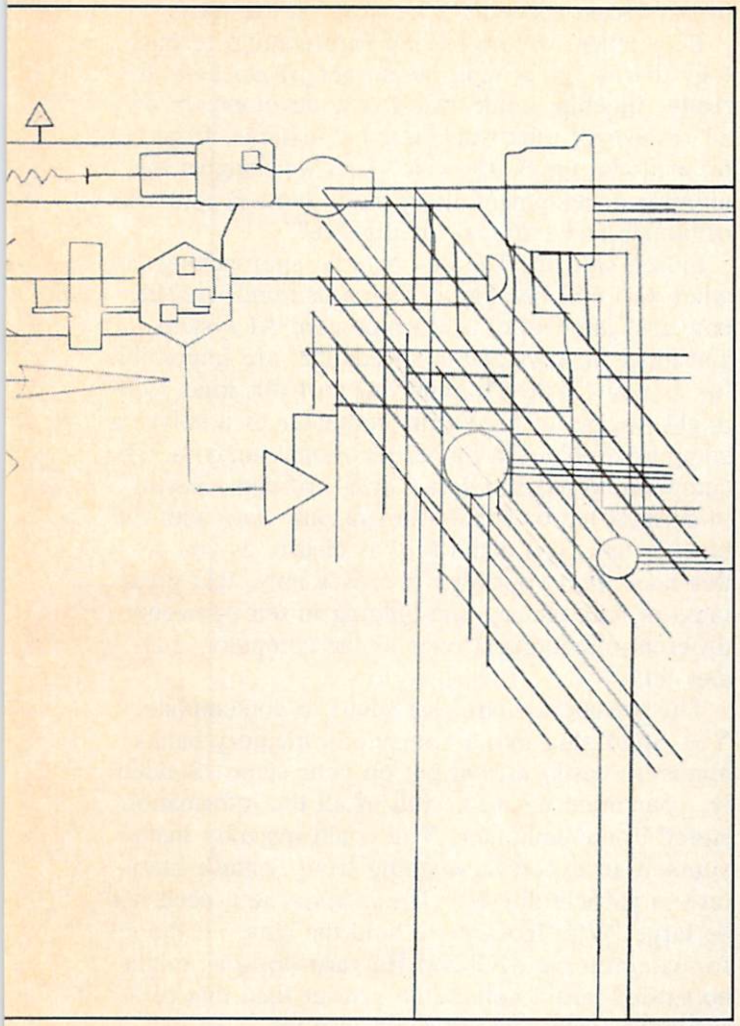
Clear a space on the kitchen counter. The counter top here represents the "base" of the biochip, a complex protein known as the "oriented anitgen monolayer." Now make up about two ounces of sticky bread dough and place a small blob of it on the counter. The dough stands for a kind of organic adhesive called a "peptide." Into the dough, insert an upright breadstick. This represents a big molecule called a "monoclonal antibody," which is produced by genetic engineering.

After making sure the breadstick is fixed firmly in the dough, add another glob of dough to the top of the stick and mount on it another, shorter breadstick. This upper piece of breadstick stands for a second monoclonal antibody.

You may have guessed by now that the two breadstick "antibodies" are supposed to be connected somehow other than with the dough "peptides"; and you are right. Take two bits of the remaining dough, put each of them on the "leaning" side of the breadsticks, and string between them a strand of cooked spaghetti about four inches long. In a real biochip, this "spaghetti" is something called a "molecular electronics switch array," a lengthy string of organic molecules that forms between peptide anchor sites on the sides of the antibodies, joining the upper and lower parts of the structure with a switching apparatus that lets the two antibodies pass signals to each other.

This carbohydrate Rube Goldberg device is almost complete. It needs only one more part—something to represent the "gate" that lets impulses into and out of the system. Using a toothpick, mount a marshmallow on top of a cherry tomato. Then set them, cherry tomato on the bottom, next to the lower breadstick, opposite the spaghetti strand. Use your last bit of leftover dough to secure the marshmallow to the side of the breadstick, so that the whole assemblage will stand by itself. The marshmallow here models an enzyme, and the tomato a metal compound. Together they form an in-and-out pathway for signals.

The actual biochip does not have to be assembled so directly by humans as our little model does. In fact, the biochip practically grows itself. Just throw in the right chemicals under the proper conditions, and nature does the rest. There is no need for big ovens to bake silicon chips, e-beams to carve patterns in them, painstakingly drawn masks to photolithograph patterns on the oxidized discs. Organic



chemistry takes care of the whole process. And the result is a circuit density far greater than anything humans could hope to achieve with conventional methods. Today a chip can hold perhaps 50,000 to 100,000 bits of information. A biochip could hold *50 billion to 100 billion*—a millionfold increase. One trillion biochips could fit on a postage stamp; a hundred trillion on a postcard. If you have trouble visualizing these figures, think of them this way: recording information on biochips, you could fit a ten-volume biography of every individual human on earth into an area about the size of your thumbnail—and still have storage capacity left over.

But compactness is only part of the marvel of biochips. A biochip could work much faster than a conventional integrated circuit. These things are as quick as they are tiny. A biochip would operate perhaps 10 billion times faster than the most advanced home computers in use today. To put it in more easily understood terms, the biochip is like a person who can do a job in one second, as opposed to his fellow worker who takes half an hour. It is easy to see which of those workers an employer would prefer to hire; it is equally easy to understand why business and industry are taking such a keen interest in biochip technology. It could make smart machines still smarter, tiny microprocessors still smaller, supercomputers yet more super.

Cost? Probably very reasonable. Nature has done most of the "design work" already, in the process of evolving our biochemistry.

Assembly would be little more than a matter of cooking up the right kind of "soup" in which the biochip components could assemble themselves. And there would be no need to chill this kind of organic computer, because biochips would not produce excessive amounts of heat. They would be powered by enzyme reactions instead of heat-producing electricity. Cool, compact, capacious, and consummately fast: that about sums up the promise of biochips.

How close are we to developing a practical biochip computer? Opinions differ here. Some conservative scientists think such a computer, if possible at all, will have to wait until well into the coming century. Other computer experts are less skeptical. They think a few hundred man-years of work—not much at all, by the standards of 1980s R&D—could produce the first working biochip components.

Biochips. We have them, in the form of nerve cells.

Computers may soon have them, in the form of ultrasmall chemical complexes. What would happen if the twain should meet?

Suppose there were some way to form a bridge between the biochips in a computer and the cells of the human brain. What if we could connect the computer, that remote annex that we built for our mind, with the very mind that gave the computer birth? Each system has been evolving in its own way, developing its own special set of expert abilities. The human brain has a vast complex of vague and mysterious but invaluable skills; the computer has the gifts of tremendous speed and all but unlimited storage capacity. What if the two could meet directly, instead of communicating through the slow and indirect media of sight and hearing? What if the computer could enter the human mind, like two persons meeting in a room? And what if the human mind could browse directly through the "mind" of a computer, like a bibliophile visiting a rare book store?

In that case, the two branches of evolution—Darwinian, as represented by our own minds, and Lamarckian, embodied by the computer—would come together in a grand synthesis—a sum that might prove much greater than all its parts.

Idle fantasy? Perhaps not. The biochip has a feature that might make a melding of machines and men a reality on the intellectual level, almost as soon as biochip computers see the light of day.

They may literally see that light, because biochips are being considered as a means to restore sight to the blind. The protein used for the base of the biochips can bind with nerve cells and, at the

same time, conduct an electrical current to and from tiny electrodes leading to sources outside the body. In this way the biochip materials could supply a direct link between the human brain and a powerful biochip computer. The human and the organic machine could then form a single system, like the natural brain and eye. A computer would view the world around it through a television camera or other optical device, convert the picture into signal form, and feed the signals right into the vision centers of a blind person's brain—thus restoring sight. This technique is taken so seriously in some quarters of the scientific community that as of this writing (1982) the National Science Foundation (NSF) is funding research on ways of "gluing" biochip proteins to neurons.

Biochips, then, could lead to artificial eyes—maybe eyes better than our natural ones. If the computer took its images from special cameras—say, infrared imagers, which convert heat emissions into pictures—then someone on the other end of the biochip link could see things as no one has ever seen them before. Imagine being able to see heat, or ultraviolet radiation, or even radioactivity, as plainly as you now can see the glow of a lightbulb. That is only one of the marvels that biochip technology may make possible in the next few years.

Biochips might also provide the so-far missing link in a fantastic technology known as "telepresence." That expression was coined by Marvin Minsky of MIT, and describes a setup in which a person could experience all the sensations of flying a plane, or driving a lunar rover across the moon, or whatever—without ever leaving a comfortable chair on the ground. Minsky imagines feeding sensory input from some distant device (say, the wingtip cameras and motion sensors of a plane in flight) by telecommunications to an operator some distance away. The operator could then fly the plane on the basis of what the relayed information told him. Only he would feel as if the aircraft *were his own body.*

Telepresence is a fascinating concept, and one can easily see how it could help to save memory, time, and lives. It could enable us to visit and work in all kinds of hostile environments without subjecting ourselves personally to risk. A telepresent person could guide a submarine along the sea floor or a tank across a battle-field, without facing the perils of abyssal pressures or shot and shell.

Telepresence would be hard to arrange, using present-day technology. But with biochips the problem of melding men with machine might be solved. Simply transmit the data from the tank or plane or whatever to a biochip computer; transfer the data, as electronic signals, from the computer to conduits leading to the brain; and the signals will flit across the protein layer between electrodes and neurons, giving the telepresent operator as good a perception of the distant scene as if he were on the spot

Remember, we are talking here about a technology that is just around the corner, if not here already. Biochips could lead to the development of all manner of man-machine combinations, from better artificial limbs to—what? Can we imagine the ultimate development along these lines—a synthesis of human and artificial intelligence?

Indeed we can. The day may be approaching fast when you will be able to join your mind with the powerful intellect of a biochip-based AI system, and think in ways and at speeds that are impossible for us now. With a "mind link" of this kind, you might be able to follow the computer as it solves a complex problem in multidimensional analysis. The computer would handle a "blob" of data existing in perhaps ten or twenty dimensions. And with the biochip link, you could see, as clearly as you see this book with your own two eyes now, that great mass of data pulsing and bulging in ten or twenty different directions at once as the computer operates on it.

The possibilities are marvelous to contemplate. You could plug into a computer's memory banks almost as easily as you put on your shoes. Suddenly, your mind would be full of all the information stored in the computer. You could instantly make yourself an expert in anything from Spanish literature to particle physics. The memory unit need not be large. With biochips to hold the data, all the information in the MIT and Harvard libraries might be stuffed into a volume no greater than that of a sandwich. All of Shakespeare in a BB-sized module. All of the known facts of chemistry in a unit no bigger than a peach. You may see devices like this before this century ends. Already biochip units, for use in tandem with the human mind, are getting such serious attention in some circles that the gadgets have a name: *transmogs,* short for *transmogrifiers.* John von Neumann would have been delighted. He once pointed out how quickly the flood of information is outpacing our ability to keep up with it, and used an illustration from his own field, mathematics. There was a time not too long ago, he said, when a mathematician could be expected to know, if not master completely, all the branches of math. Now our mathematical knowledge is expanding so fast that even an expert in mathematics, devoting a lifetime of study to it, could reasonably expect to know only about ten percent of it all, at the very most. As long as we depend on the crude input systems of sight and hearing, and the limited storage capacity of our own natural brains, that ten percent figure is likely to keep dropping. But with transmogs to store information for us, and biochip "interfaces" to help feed it into our minds, we might reverse that trend—and start pushing that ten percent figure back up toward one hundred. Biochips could give us a life preserver, so to speak,

that would allow us to keep our heads above the flood of new information.

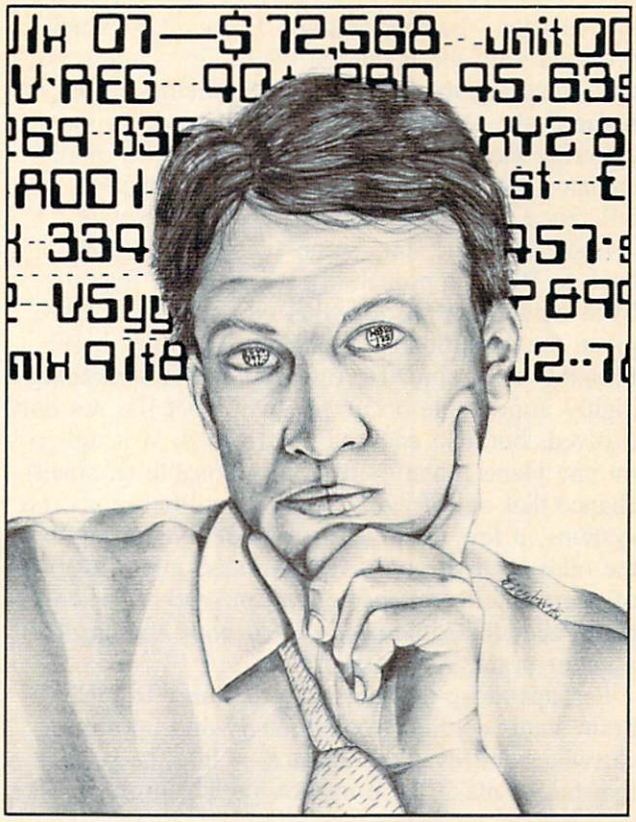
Information. Ultimately, everything is information. You, as an organism and a personality, are made up of information. Society is made up of information. Economics, technology, resource allocation—all of them are matters of information. And all the problems of our world, from pollution to overpopulation to famine to war, exist because our information supply is inadequate.

With that thought in mind, take a walk through the Berkeley campus. This campus is where the Speech movement gave rise to the campus upheavals of the sixties, and at Berkeley the students still speak out, through handbills and posters and graffiti, on every problem under the sun. All the horsemen of the Apocalypse are discussed on Berkeley's walls and bulletin boards, plus a few more recent specters brought in to swell the cavalry of doom: nuclear waste, mind control, and so forth. How are we to deal with the woes that face our world? The Berkeleyites have plenty of suggestions. "'Global socialism,'' one handbill advises. ''Laissez-faire,'' says another only a few inches away on a bulletin board. Throw a rock in Berkeley, and chances are it will hit someone's slogan for saving society. Some are intriguing. Some are downright weird. And some are simply naive. On a wall near the gym, some zealot with a spray can has scrawled, "THE WORLD NEEDS JESUS!"

What the world *really* needs is a better way of handling information, because information is all-powerful (a fact which the Bible, interestingly, acknowledges when it describes the Almighty God in terms of information units: "In the beginning was the Word, and the Word was with God, and the Word was God."). Without information, nothing can happen. But *with* the right information, virtually anything is possible. And by coming up with the proper information, one can turn want and war into peace and plenty. The question is, how and where to look for the information we so desperately need?

There are two places to search, because there are two kinds of knowledge: know-what and know-how. Know-what consists of all the little individual bits of information kept on record—names, dates, measurements, and so forth. Know-how is an understanding of how to apply that knowledge to practical tasks.

Neither kind of information is much good without the other. Take the case of electromagnetism. Up to the nineteenth century, science had accumulated a lot of know-what knowledge about the phenomenon of electricity. But that knowledge was largely disorganized. We were unsure just how electromagnetism behaved, because we were ignorant of the laws that govern its workings. Then James Clerk Maxwell supplied the crucial bit of know-how: a set of equations that described elec-



tromagnetism perfectly. Maxwell's equations made it possible for us to master electricity, and the results of his know-how contribution are all around you, from the telephone in your bedroom to the spark plugs in your car. Satellite communications, electronic fund transfers, electric-eye doors—these and millions of other advances all sprang ultimately from that one piece of added know-how. This is what information can do.

Most likely the information that would cure cancer and solve our energy woes is sitting on a shelf somewhere right now, waiting to be pulled out and put to use. If we could only assemble all that know-what information, plug it into the necessary know-how, and put it to work, the result might be a golden age of peace and prosperity for all the peoples of the world. Is there any way to join all of our know-what knowledge with all of our know-how?

Perhaps there is.

If we can fuse computers with the human mind, through biochip technology, then these two great bodies of knowledge will come together in a single man-machine system—a binary brain. The computer will provide virtually endless and infallible memory, plus prodigious powers of data-crunching. The human brain and mind will supply all that special know-how that a human acquires, both as a being in a physical body and as a part of society. Each part of the system will give the other something it desperately needs but lacks. And the two avenues

of evolution—Lamarckian and Darwinian—which diverged with the invention of the first computers, will come together again, like the themes of a fugue at the close. The results will be more awe-inspiring than we can imagine. We can no more envision the deepest workings of such an intelligence, than a dog can understand the stars.

Look up at the stars some evening, and think back to Pascal's probability theory. Had Pascal lived on another world, what odds would he have given the evolution of intelligent life down here? Probably very slim. Here on our world something highly improbable occurred. Intelligent life not only evolved, but also endured and thrived. If it did so on one planet, then there is a reasonable (if small) chance that out of every hundred million or so star systems, a few intelligent forms of life appear, beat the odds, and survive. And perhaps, in the history of every intelligent species, a time arrives when society faces the same need we do now: the need for a better brain. A binary brain.

Perhaps those that do make the jump to binary brain status continue to live, and wind up turning cartwheels all over the cosmos—while the species that fail to make the transition perish, drowned like rats in a flood of their own information. And perhaps one of the successful species is out there right now, staring us in the face—but is so far advanced over us intellectually that we fail to recognize its intelligence, just as the work going on at Berkeley's physics labs makes no impression on the butterflies in the Berkeley hills.

From those hills, the view is magnificent. Across the bay rise the towers of San Francisco. To the right are the Marin hills, joined to the city by the bright orange span of the Golden Gate Bridge.

This gateway truly was golden, for it opened onto the realm of the vast Pacific, and from that realm America reaped wealth and power beyond its dreams. Now another golden gateway appears to be opening—this time for the whole human species, and to a whole universe rather than a mere ocean—as biology and technology prepare us, perhaps for the next step in our own evolution.

But we have to take that step ourselves. The decision is ours. And only time will tell whether we walk through that portal toward binary brain status, and face the future with minds made new—or turn away and be content, like the Sphinx and the brontosaurus, to live as little minds on a little world.□

Dare To Join A Users Group

By Pete Lobl

In 1977 Commodore, a little-known watch company, decided to market a personal computer shipped ready to run with monitor and BASIC built in. The 2001, as it was called, elicited great excitement from people around the country and abroad. Along with the increasing numbers of users came hundreds of dealerships—800 as of a year ago—to provide demonstrations of new products and to help the user with his problems. This "user-friendly" system was fine for all involved, but as 1983 came along things began to change drastically.

Following the lead of videogame systems, home computers began to find their way into the average household. No longer were computers only for hackers and businessmen. The VIC-20 was down to $179 and the 64 was being discounted to $500—prices at which the average person could afford to own a computer.

But while the number of non-technical users rose astoundingly, the amount of support did not. In fact, within a year, the number of Commodore dealers in the United States was cut by more than half. Why, you ask? Let's look at the main reason.

At the same time that VIC-20 sales began to explode, Commodore made a fateful marketing decision: to allow mass-merchandisers such as Toys-R-Us and K-Mart to carry the VIC and 64 line. This move worried dealers, but not nearly as much as it should have. Toys-R-Us and others bought in massive quantities, allowing Commodore to give them super-discounted rates. This allowed the mass merchandisers to offer supplies to the public at unheard-of prices. I think it's obvious why so many dealers folded: how can you compete with a place that charges 20% less than you and has an advertising budget equal to your phone number?

Thus, though Commodore is today churning out more computers than ever before, its user support is at an all-time low. For the average consumer, a computer is a new and confusing device. Whatever application you purchased it for, you will undoubtedly come up with what seems like a zillion questions that have no answer. Where can you turn to obtain this golden knowledge?

To Commodore and their user hotline? *HA!!* From my experience, that's nothing but a waste of time and money. Don't take my word for it. If you don't mind making a long-distance call to Pennsylvania, dial (215)555-1212 and ask the operator for Commodore's User-Service number. The operator will ask you where Commodore is located; tell her or him Westchester. Dial the number the operator gives you, and when you're asked for your problem, give one that really baffles you. Odds are that the person on the other end will be as baffled as you.

So, where can you turn for support, knowledge, and honest opinions? Look no further than a users group.

Why is a users group for you? I've already men-

tioned the need for general information on any topic. In almost every group, there is at least one person knowledgeable enough to make you feel comfortable that your questions will be answered.

Another good reason to join a group is to get first-hand software and hardware reviews. Want an opinion on that new printer from someone other than the salesman who wants to send you home with it, or the magazine that, in addition to a review of the product, is carrying $10,000 worth of advertising for it? Ask the members of your group. They may not have all the answers you need, but generally the information you gain will help.

The final reason to join a users group concerns the most important task of a computer owner: acquiring the software he needs to run it. Most groups have a library of public-domain software which is free for you to copy. Some groups promote the illegal practice of copying copyrighted programs. Without trying to preach right or wrong on this topic, all I will say is that the copying of sold software is widespread—both within and without the groups. (Remember that every time you copy a program, a company loses profits and may have to raise prices to make it up. It's a two-way street.) In any case, you will be able to learn more about the currently available software and may discover a program that you have to get, even if it means buying it!

Overall, you can see that the users group offers you far more than you can obtain from any other source. Some groups are more formal than others; some require dues, some produce newsletters. In any event, you can't lose by attending a meeting near you. There's no reason to suffer any longer! .



Here in Long Island, NY, the many local users groups are trying to unite and form one mass organization. We are attempting to extend this network across the country and abroad. If you'd be interested in joining this group (as a member of another group or without affiliation), send your name and address to:

International Commodore Society
c/o AHOY!
45 West 34th Street—Suite 407
New York, NY 10001

Send any comments or questions you may have on this new group along with your address. We will get back to you with more info as soon as possible.

The group is not affiliated with CBM in any way, and hopefully it will stay that way. I hope to be hearing from you!

RELATIVE FILES

Continued from page 28

cussed here. It is similar to the program listed in our article on sequential files in the last issue of *AHOY!* Notice how much more convenient relative files are for this application.

Next issue, we will discuss the *random* file type. Stay tuned!

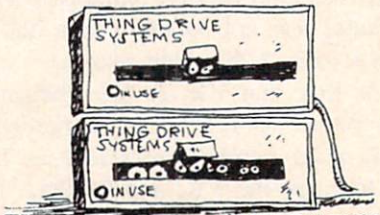
SEE PROGRAM LISTING ON PAGE 91

Readers are invited to submit any additional information on the disk drive and the DOS to the author in care of this magazine. We cannot guarantee an individual response, but will incorporate the most interesting comments into our follow-up article.

GUIDED TOUR

Continued from page 38

write operation or turning the drive off and back on. Reading the error channel does not seem to clear up this problem.



PROGRAM LISTINGS

On the following pages, there are listed several programs that you might wish to try out on your own computer. Before you do so, there are a few things that you will need to know.

Certain computer commands are displayed on the monitor by various odd looking characters. To get your computer to *print* these command symbols (rather than perform the action of the command) you need to enter the quote mode. To do so, hold down the SHIFT key and press the "2" key; a pair of quotation marks will appear. This tells the computer that the next symbol is to be represented by a character. To get out of the quote mode just type in another set of quotation marks. You will also enter the quote mode when you INserT spaces or characters into a line. The easiest way to get out of the quote mode is to hit the RETURN key.

In Ahoy's program listings, you will run into letters and/or numbers surrounded by a pair of brackets. You will notice that these brackets appear neither on your keyboard nor in your printed programs. You, in all probability, use a dot matrix printer, but for the purpose of reproduction, we at ***AHOY!*** use a letter quality printer that is incapable of reproducing the command symbols.

Thus, when you are in the quote mode and press the SHIFT and CLR/HOME keys at the same time, your screen (and dot martix printer) would indicate this command with a character that looks like a heart ( ). Since a letter quality printer is unable to duplicate this symbol it substitutes an alternate code that is listed within the brackets{ } In the case of the SHIFT CLR/HOME symbol, our printer will print {SC}. What this command does is tell the computer to clear the screen, and return the cursor to the "home" position (top left of the screen).

An alternate way of entering the various commands listed below, as well as the several other graphic symbols and characters, is by typing in their appropriate character strings (CHR$). For example, the CLR/HOME command is CHR$(147). While typing character strings requires a few extra strokes, it does facilitate reading in the printed version, or when editing programs. For a complete list of CHR$ codes, consult the appendix in the back of your user manual

Below are listed a series of commands, the character that represents them on the screen or dot matrix printer, and how they appear in an ***AHOY!*** program listing.

When You See	It Means	You Type	You Will See
{SC}	Screen Clear	SHIFT CLR/HOME	
{HM}	Home	CLR/HOME	
{CU}	Cursor Up	SHIFT ↑ CRSR ↓	
{CD}	Cursor Down	↑ CRSR ↓	
{CL}	Cursor Left	SHIFT ← CRSR →	
{CR}	Cursor Right	← CRSR →	
{SS}	Shifted Space	SHIFT space	
{IN}	Insert	INST	
{RV}	Reverse On	CNTRL 9	
{RO}	Reverse Off	CNTRL 0	
{BK}	Black	CNTRL 1	
{WH}	White	CNTRL 2	
{RD}	Red	CNTRL 3	
{CY}	Cyan	CNTRL 4	
{PU}	Purple	CNTRL 5	
{GN}	Green	CNTRL 6	
{BL}	Blue	CNTRL 7	

When You See	It Means	You Type	You Will See
{YL}	Yellow	CNTRL 8	
{OR}	Orange	COMMODORE 1	
{BR}	Brown	COMMODORE 2	
{LR}	Light Red	COMMODORE 3	
{G1}	Grey 1	COMMODORE 4	
{G2}	Grey 2	COMMODORE 5	
{LG}	Light Green	COMMODORE 6	
{LB}	Light Blue	COMMODORE 7	
{G3}	Grey 3	COMMODORE 8	
{F1}	Function 1	F 1	
{F2}	Function 2	F 2	
{F3}	Function 3	F 3	
{F4}	Function 4	F 4	
{F5}	Function 5	F 5	
{F6}	Function 6	F 6	
{F7}	Function 7	F 7	
{F8}	Function 8	F 8	

CD = CURSOR DOWN

CR = CURSOR RIGHT

NIGHT ATTACK

From page 26

PROGRAM 1

```
1 POKE36879,8:PRINT"{SC}"
2 PRINT"{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CR}{CR}{CR}{CR}{CR}NIGHT ATTACK"
3 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CR}{CR}{CR}{CR}{CR}{CR}{CR}CREATED BY,"
4 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CR}{CR}{CR}{CR}J.R. ALONSO, JR."
5 V=V+1:IFV/2=INT(V/2)THENPRINT"{RD}":GOTO4
6 IFV=251THEN8
7 PRINT"{BL}":GOTO4
8 PRINT"{SC}       {RV}INSTRUCTIONS{RO}"
9 PRINT"{CD}{RV}1.{RO}JOYSTICK OR          KEYBOARD       J"
10 PRINT"                    "
11 PRINT"       BLAST      N    M       {RV}F7{RO}"
12 PRINT"                    {CU}]"
13 PRINT"                 {RV}SPACE{RO} "
14 PRINT"{RV}2.{RO}SHOOT ALIENS BEFORE"
15 PRINT"  THEY DESTROY NOVA "
16 PRINT"  CITY!!!":PRINT"{CD} LINE UP CROSSHAIR"
17 PRINT"  ON ALIENS AND FIRE!":PRINT
18 PRINT"{RV}3.{RO}YOU GET TEN POINTS":PRINT"  PER HIT AND TWENTY":PRINT"  FIVE PER DIRECT HIT"
19 PRINT"  NEW WAVE EVERY 500":PRINT"  POINTS!!!":PRINT"{CU}{CU}"
20 FORX=7168TO7679:POKEX,PEEK(X+25600):NEXT
25 FORX=7312TO7319:POKEX,PEEK(X+25520):NEXTX
30 FORX=7168TO7175:READA:POKEX,A:POKEX+120,A:NEXT
35 FORX=7176TO7247:POKEX,0:NEXT:FORX=7280TO7287:POKEX,0:NEXT
40 FORX=7248TO7279:READA:POKEX,A:NEXT
45 FORX=7384TO7423:READA:POKEX,A:NEXT
50 FORX1=1TO5:READB:FORX=1TO8:READA:POKEB,A:B=B+1:NEXTX:NEXTX1
55 FORTD=1TO400:NEXT
75 PRINT"{SC}{CD}{CR}BY, J.R.ALONSO,JR."
80 POKE631,131:POKE198,1:END
100 DATA129,129,219,255,255,195,129,129
110 DATA102,60,90,255,102,255,36,102
120 DATA73,54,34,65,34,20,28,99
130 DATA60,102,129,255,36,36,165,195
140 DATA24,66,36,255,0,129,90,60
150 DATA165,129,165,129,165,129,165,255
160 DATA240,144,144,144,159,145,241,255
170 DATA0,0,0,0,255,129,165,129
180 DATA92,68,92,68,92,68,92,126
190 DATA24,24,60,36,36,36,126,68
200 DATA7432,0,0,0,0,24,60,126,68
210 DATA7456,9,218,122,127,63,255,255,255
220 DATA7472,0,0,128,200,240,255,255,255
230 DATA7480,33,51,123,255,255,255,255,255
240 DATA7512,8,8,8,127,8,8,8,0
```

PROGRAM 2

```
5 POKE36879,8:PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}PRESS K IF USING THE  KEYBOARD."
6 POKE52,28:POKE56,28:CLR:GETK$:IFK$=""THEN6
10 CU=36869:CC=30720:SC=7702:L=7932:SE=8185:S=36878:S1=S-2:S2=S-1:JR=37154:LE=25
15 DEFFNB(E)=INT(RND(1)*E)+7680:DEFFNC(E)=INT(RND(1)*E)+21
20 GOSUB300:H=.:MS=15:GOSUB860
35 Z4=FNB(21)-22:Z5=INT((Z4-SC)/2)+SC:Z6=INT((Z4-Z5)/3)+SC
50 IFK$="K"THENGOSUB750:GOSUB400:POKES,10:POKES1,240:GOTO135
55 GOSUB400:GOSUB810:POKES,10:POKES1,240:IFJ0THEND=D+2
75 IFJ1THEND=D+44
80 IFJ2THEND=D-2
85 IFJ3THEND=D-44
```

```
•90 IFFBTHENGOSUB600:GOSUB500
•135 POKEL,32:IFL+D<SCTHEND=.
•140 IFL+D>8120THEND=.
•145 L=L+D:POKEL+CC,7:POKEL,43:POK
ES,.:D=.:IFH>45THEN700
•200 POKEZ4,32:Z4=Z4+FNC(3):POKEZ4
,.:POKEZ4+CC,2:IFZ4>SETHENZ4=7712
:H=H+1
•210 POKEZ5,32:Z5=Z5+FNC(3):POKEZ5
,.:POKEZ5+CC,3:IFZ5>SETHENZ5=7705
:H=H+1
•220 POKEZ6,32:Z6=Z6+FNC(3):POKEZ6
,.:POKEZ6+CC,5:IFZ6>SETHENZ6=7719
:H=H+1
•230 GOTO50
•300 POKECU,255:POKECU+10,8:PRINT"
{SC}":FORX=1TO22:PRINT:NEXT:PRINT
"{YL}]! ← ] !←] !]←  ]]← ]{CL}{C
L}{CL}"
•310 PRINT"{CU}{YL}[↑£↑£[£↑£↑[£↑[↑
££[[↑£[{CR}{CR}{HM}":PRINT"{HM}{C
D}{CR}{IN}":POKE218,158:H=.:RETUR
N
•400 PRINT"{HM}{BL}S:";PO:PRINT"{H
M}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR
}{CR}{CR}RS:";HS:IFPO>MLTHENGOSUB
850
•410 RETURN
•500 PT=PO:IFPEEK(L)=.THENPO=PO+25
:POKES,10:POKES1,208:FORDE=1TO10:
NEXTDE:X=L:GOTO530
•510 FORX=-23TO23:IFPEEK(L+X)=.THE
NPO=PO+10:X=L+X:GOTO530
•515 IFX=-21THENX=-2
•520 IFX=1THENX=20
•525 NEXTX:GOTO545
•530 IFZ4=XTHENPOKEZ4,32:Z4=7690:G
OTO545
•535 IFZ5=XTHENPOKEZ5,32:Z5=7690:G
OTO545
•540 POKEZ6,32:Z6=7690
•545 IFPO>PTTHENGOSUB550:GOSUB400:
RETURN
•547 GOSUB590:RETURN
•550 L=L-23:FORX1=0TO46:N=N+1:POKE
L+X1,N:POKEL+X1+CC,2:IFX1=2THENX1
=21
•555 IFX1=24THENX1=43
•560 NEXTX1:N=0:FORX1=1TO74:READA,
B:A=A+7168:POKEA,B:NEXTX1:RESTORE
•565 FORX1=0TO46:POKEL+X1,32:IFX1=
2THENX1=21
•570 IFX1=24THENX1=43
•575 NEXTX1:L=L+23
•580 FORX=7176TO7248:POKEX,0:NEXT
•590 POKES2,222:FORY1=15TO0STEP-1
•595 POKES,Y1:FORZ1=1TOLE:NEXTZ1,Y
1
•600 POKES,.:POKES1,.:POKES2,.:RET
URN
•700 GOSUB600:PRINT"{SC}":PRINT"{C
D}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD
}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{YL}"
•705 PRINT"'$&'$&&''$$&$'&'$&''$&"
:PRINT"{RD}$$&&'$&''&&''$$$&''&$$
{CU}"
•708 LE=125:GOSUB590
•730 Z9=FRE(X):IFPO>HSTHENHS=PO
•740 H=.:PO=.:RESTORE:GOTO10
•750 K=PEEK(197):IFK=36THEND=D+2
•760 IFK=32THEND=D+44
•770 IFK=28THEND=D-2
•780 IFK=20THEND=D-44
•790 IFK=63THENGOSUB600:GOSUB500
•800 RETURN
•810 POKEJR,127:P=PEEK(JR-2)AND128
:J0=-(P=.):POKEJR,255:P=PEEK(JR-3
):J1=-((PAND8)=.)
•830 J2=-((PAND16)=.):J3=-((PAND4)
=.):FB=-((PAND32)=.):RETURN
•850 FORX=175TO215:POKES1,X:FORX1=
15TO10STEP-1:POKES,X1:NEXTX1:NEXT
X
•860 FORCH=7168TO7175:POKECH,PEEK(
CH+(8*MS)):NEXTCH:MS=MS+1:IFMS=16
THENMS=10
•870 ML=ML+500:RETURN
•880 DATA44,16,43,24,45,52,44,56,4
2,52,46,82,44,124,41,82,47,145,44
,255,40,145,56,1
•885 DATA64,16,72,128,15,1,36,1,52
,128,23,16,31,128,57,2,65,16,36,3
,52,192,73,64
•890 DATA14,2,22,16,30,64,36,7,52,
224,58,4,66,16,74,32,13,4,36,15,2
1,16,29,32
•895 DATA59,8,67,16,36,31,52,248,7
5,16,12,8,20,16,28,16,36,63,52,25
2,60,16,68,16
•900 DATA76,8,11,16,36,127,52,252,
19,16,27,8,61,32,36,255,52,255,69
,16,77,4,10,32
•905 DATA18,16,26,4,62,64,70,16,78
,2,9,64,17,16,25,2,63,128,71,16,7
9,1,8,128,16,16,24,1
```

MULTI-DRAW 64

From page 24

PLOT 0

```
1000 ; THIS IS THE PLOT 0 ROUTINE
1010 ; IT IS FOR MULTI-COLORED BI
T
1020 ; PATTERN 00
1030 ; DONE MAY 22,1983
1040 ;
1050 ; PROGRAM STORES VAR'S IN CA
SSETTE
1060 ; BUFFER. COLOR 0 IS AT $D02
0
1070 ;
1080 ; OR TABLE IS AT $0342-$0349
1090 ; AND TABLE IS AT $0354-$035
B
1100 ;
1110 *=$0B00
1120 XCO = $033C
1130 YCO = XCO+1
1140 CHAR = YCO+1
1150 ROW = CHAR+1
1160 LINE = ROW+1
1170 BITT = LINE+1
1180 PLOT = $F9
1190 IND0 = PLOT+4
1200 COLOR0 = $D021
1210 TABLE = $0342
1220 TABLE2 = $0354
1230  LDA XCO    ; X COORDINATE
1240  LSR A      ; HAS TO BE 0-127
1250  LSR A
1260  LSR A
1270  STA CHAR
1280  LDA YCO    ; Y COORDINATE
1290  LSR A      ; HAS TO BE 0-199
1300  LSR A
1310  LSR A
1320  STA ROW
1330  LDA YCO
1340  AND #7
1350  STA LINE
1360  LDA XCO
1370  AND #7
1380  STA BITT
1390  LDA #7
1400  SEC
1410  SBC BITT
1420  STA BITT
1430  LDX ROW
1440  CLC
1450  BEQ PATCH
1460 BEGIN INC PLOT+1 ; ADD 256
1470  LDA PLOT
1480  ADC #64      ; ADD 64 MORE
1490  STA PLOT
1500  BCC NEXT
1510  INC PLOT+1
1520  CLC
1530 NEXT DEX
1540  BNE BEGIN
1550 PATCH LDX CHAR   ; PLOT=BASE
+ROW*320
1560  BEQ PATCH2
1570 BEGIN2 LDA PLOT
1580  ADC #8      ; ADD 8
1590  STA PLOT
1600  BCC NEXT2
1610  INC PLOT+1
1620  CLC
1630 NEXT2 DEX
1640  BNE BEGIN2
1650 PATCH2 LDA PLOT   ; PLOT=PLO
T+CHAR*8
1660  ADC LINE   ; ADD 0-7
1670  STA PLOT
1680  BCC NEXT3
1690  INC PLOT+1
1700  CLC
1710 NEXT3 LDX BITT ; PLOT=BYTE P
LOT
1720  LDY #0
1730  LDA TABLE2,X   ; LOOK UP BI
T
1740  AND (PLOT),Y
1750  STA (PLOT),Y  ; PLOT POINT
1
1760  DEX
1770  LDA TABLE2,X   ; LOOK UP BI
T 2
1780  AND (PLOT),Y
1790  STA (PLOT),Y  ; PLOT POINT
2
1800 ; HERE COMES THE COLOR PLOT
1810  RTS ; OOPS, WHEN YOU ERASE
YOU
1820 ;       DON'T COLOR !!!!
1830 ;
1840 ; PATCHES ARE FOR SPECIAL CA
SES
1850 ; WHICH I HADN'T EXPECTED.
1860 ; THEY SIMPLY ALLOW A ZERO (
```

```
0)
•1870 ; VALUE TO BE SKIPPED OVER
•1880 ; WHICH ALLOWS FOR ACCURACY
•1890 ; IN THE TOP 8 PIXELS AND IN
 THE
•1900 ; COLUMN AT THE EXTREME LEFT
 .
```

PLOT 1

```
•1000 ; THIS IS THE PLOT 1 ROUTINE
•1010 ; IT IS FOR MULTI-COLORED BI
 T
•1020 ; PATTERN 01
•1030 ; DONE MAY 22,1983
•1040 ;
•1050 ; PROGRAM STORES VAR'S IN CA
 SSETTE
•1060 ; BUFFER. COLOR 1 IS AT $035
 1
•1070 ;
•1080 ; OR TABLE IS AT $0342-$0349
•1090 ; AND TABLE IS AT $0354-$035
 B
•1100 ;
•1110 *=$0A00
•1120 XCO = $033C
•1130 YCO = XCO+1
•1140 CHAR = YCO+1
•1150 ROW = CHAR+1
•1160 LINE = ROW+1
•1170 BITT = LINE+1
•1180 PLOT = $F9
•1190 IND1 = PLOT+4
•1200 COLOR1 = $0351
•1210 TABLE = $0342
•1220 TABLE2 = $0354
•1230  LDA XCO    ; X COORDINATE
•1240  LSR A      ; HAS TO BE 0-127
•1250  LSR A
•1260  LSR A
•1270  STA CHAR
•1280  LDA YCO    ; Y COORDINATE
•1290  LSR A      ; HAS TO BE 0-199
•1300  LSR A
•1310  LSR A
•1320  STA ROW
•1330  LDA YCO
•1340  AND #7
•1350  STA LINE
•1360  LDA XCO
•1370  AND #7
•1380  STA BITT
•1390  LDA #7
•1400  SEC
•1410  SBC BITT
•1420  STA BITT
•1430  LDX ROW
•1440  CLC
•1450  BEQ PATCH
•1460 BEGIN INC PLOT+1 ; ADD 256
•1470  LDA PLOT
•1480  ADC #64      ; ADD 64 MORE
•1490  STA PLOT
•1500  BCC NEXT
•1510  INC PLOT+1
•1520  CLC
•1530 NEXT DEX
•1540  BNE BEGIN
•1550 PATCH LDX CHAR    ; PLOT=BASE
 +ROW*320
•1560  BEQ PATCH2
•1570 BEGIN2 LDA PLOT
•1580  ADC #8      ; ADD 8
•1590  STA PLOT
•1600  BCC NEXT2
•1610  INC PLOT+1
•1620  CLC
•1630 NEXT2 DEX
•1640  BNE BEGIN2
•1650 PATCH2 LDA PLOT    ; PLOT=PLO
 T+CHAR*8
•1660  ADC LINE    ; ADD 0-7
•1670  STA PLOT
•1680  BCC NEXT3
•1690  INC PLOT+1
•1700  CLC
•1710 NEXT3 LDX BITT ; PLOT=BYTE P
 LOT
•1720  LDY #0
•1730  LDA TABLE2,X    ; LOOK UP BI
 T
•1740  AND (PLOT),Y
•1750  STA (PLOT),Y   ; PLOT POINT
 1
•1760  DEX
•1770  LDA TABLE,X    ; LOOK UP BIT
 2
•1780  ORA (PLOT),Y
•1790  STA (PLOT),Y   ; PLOT POINT
 2
•1800 ; HERE COMES THE COLOR PLOT
•1810  LDX ROW
•1820  CLC
•1830  BEQ PATCH3
•1840 CONT LDA IND1
•1850  ADC #40        ; ADD 40 (1 R
```

```
OW)
•1860  STA IND1       ; FOR EACH VA
LUE
•1870  BCC NEXT4      ; OF X
•1880  INC IND1+1
•1890  CLC
•1900 NEXT4 DEX
•1910  BNE CONT
•1920 PATCH3 LDA IND1
•1930  ADC CHAR      ; ADD 0-39
•1940  STA IND1
•1950  BCC CONT2
•1960  INC IND1+1
•1970 CONT2 LDA COLOR1
•1980  AND #15
•1990  LSR A
•2000  LSR A
•2010  LSR A
•2020  LSR A
•2030  STA CHAR
•2040  LDA (IND1),Y
•2050  AND #15
•2060  ORA CHAR
•2070  STA (IND1),Y
•2080  RTS           ; COLOR IS PLO
TTED
•2090 ;
•2100 ; PATCHES ARE FOR SPECIAL CA
SES
•2110 ; WHICH I HADN'T EXPECTED.
•2120 ; THEY SIMPLY ALLOW A ZERO (
0)
•2130 ; VALUE TO BE SKIPPED OVER
•2140 ; WHICH ALLOWS FOR ACCURACY
•2150 ; IN THE TOP 8 PIXELS AND IN
 THE
•2160 ; COLUMN AT THE EXTREME LEFT
.
```

PLOT 2

```
•1000 ; THIS IS THE PLOT 2 ROUTINE
•1010 ; IT IS FOR MULTI-COLORED BI
T
•1020 ; PATTERN 10
•1030 ; DONE MAY 22,1983
•1040 ;
•1050 ; PROGRAM STORES VAR'S IN CA
SSETTE
•1060 ; BUFFER. COLOR 2 IS AT $035
2
•1070 ;
•1080 ; OR TABLE IS AT $0342-$0349
•1090 ; AND TABLE IS AT $0354-$035
B
•1100 ;
•1110 *=$0900
•1120 XCO = $033C
•1130 YCO = XCO+1
•1140 CHAR = YCO+1
•1150 ROW = CHAR+1
•1160 LINE = ROW+1
•1170 BITT = LINE+1
•1180 PLOT = $F9
•1190 IND2 = PLOT+4
•1200 COLOR2 = $0352
•1210 TABLE = $0342
•1220 TABLE2 = $0354
•1230  LDA XCO    ; X COORDINATE
•1240  LSR A      ; HAS TO BE 0-127
•1250  LSR A
•1260  LSR A
•1270  STA CHAR
•1280  LDA YCO    ; Y COORDINATE
•1290  LSR A      ; HAS TO BE 0-199
•1300  LSR A
•1310  LSR A
•1320  STA ROW
•1330  LDA YCO
•1340  AND #7
•1350  STA LINE
•1360 LDA XCO
•1370  AND #7
•1380  STA BITT
•1390  LDA #7
•1400  SEC
•1410  SBC BITT
•1420  STA BITT
•1430  LDX ROW
•1440  CLC
•1450  BEQ PATCH
•1460 BEGIN INC PLOT+1 ; ADD 256
•1470  LDA PLOT
•1480  ADC #64      ; ADD 64 MORE
•1490  STA PLOT
•1500  BCC NEXT
•1510  INC PLOT+1
•1520  CLC
•1530 NEXT DEX
•1540  BNE BEGIN
•1550 PATCH LDX CHAR   ; PLOT=BASE
+ROW*320
•1560  BEQ PATCH2
•1570 BEGIN2 LDA PLOT
•1580  ADC #8      ; ADD 8
•1590  STA PLOT
•1600  BCC NEXT2
```

```
•1610  INC PLOT+1
•1620  CLC
•1630 NEXT2 DEX
•1640  BNE BEGIN2
•1650 PATCH2 LDA PLOT   ; PLOT=PLO
T+CHAR*8
•1660  ADC LINE   ; ADD 0-7
•1670  STA PLOT
•1680  BCC NEXT3
•1690  INC PLOT+1
•1700  CLC
•1710 NEXT3 LDX BITT ; PLOT=BYTE P
LOT
•1720  LDY #0
•1730  LDA TABLE2,X   ; LOOK UP BI
T
•1740  AND (PLOT),Y
•1750  STA (PLOT),Y  ; PLOT POINT
 1
•1760  DEX
•1770  LDA TABLE,X   ; LOOK UP BIT
 2
•1780  ORA (PLOT),Y
•1790  STA (PLOT),Y  ; PLOT POINT
 2
•1800 ; HERE COMES THE COLOR PLOT
•1810  LDX ROW
•1820  CLC
•1830  BEQ PATCH3
•1840 CONT LDA IND2
•1850  ADC #40       ; ADD 40 (1 R
OW)
•1860  STA IND2      ; FOR EACH VA
LUE
•1870  BCC NEXT4     ; OF X
•1880  INC IND2+1
•1890  CLC
•1900 NEXT4 DEX
•1910  BNE CONT
•1920 PATCH3 LDA IND2
•1930  ADC CHAR     ; ADD 0-39
•1940  STA IND2
•1950  BCC CONT2
•1960  INC IND2+1
•1970 CONT2 LDA COLOR2
•1980  AND #15
•1990  LSR A
•2000  LSR A
•2010  LSR A
•2020  LSR A
•2030  STA CHAR
•2040  LDA (IND2),Y
•2050  AND #15
•2060  ORA CHAR
•2070  STA (IND2),Y
•2080  RTS          ; COLOR IS PLO
TTED
•2090 ;
•2100 ; PATCHES ARE FOR SPECIAL CA
SES
•2110 ; WHICH I HADN'T EXPECTED.
•2120 ; THEY SIMPLY ALLOW A ZERO (
0)
•2130 ; VALUE TO BE SKIPPED OVER
•2140 ; WHICH ALLOWS FOR ACCURACY
•2150 ; IN THE TOP 8 PIXELS AND IN
 THE
•2160 ; COLUMN AT THE EXTREME LEFT
.
```

PLOT 3

```
•1000 ; THIS IS THE PLOT 3 ROUTINE
•1010 ; IT IS FOR MULTI-COLORED BI
T
•1020 ; PATTERN 11
•1030 ; DONE MAY 22,1983
•1040 ;
•1050 ; PROGRAM STORES VAR'S IN CA
SSETTE
•1060 ; BUFFER. COLOR 3 IS AT $035
3
•1070 ;
•1080 ; OR TABLE IS AT $0342-$0349
•1090 ;
•1100 *=2048       ; PROGRAM STARTS
 WHERE
•1110 XCO = $033C ; BASIC BEGINS N
ORMALLY
•1120 YCO = XCO+1
•1130 CHAR = YCO+1
•1140 ROW = CHAR+1
•1150 LINE = ROW+1
•1160 BITT = LINE+1
•1170 PLOT = $F9
•1180 IND3 = PLOT+2
•1190 COLOR3 = $0353
•1200 TABLE = $0342
•1210  LDA XCO    ; X COORDINATE
•1220  LSR A      ; HAS TO BE 0-255
•1230  LSR A
•1240  LSR A
•1250  STA CHAR
•1260  LDA YCO    ; Y COORDINATE
•1270  LSR A      ; HAS TO BE 0-199
•1280  LSR A
•1290  LSR A
```

```
•1300  STA ROW
•1310  LDA YCO
•1320  AND #7
•1330  STA LINE
•1340  LDA XCO
•1350  AND #7
•1360  STA BITT
•1370  LDA #7
•1380  SEC
•1390  SBC BITT
•1400  STA BITT
•1410  LDX ROW
•1420  CLC
•1430  BEQ PATCH
•1440 BEGIN INC PLOT+1 ; ADD 256
•1450  LDA PLOT
•1460  ADC #64      ; ADD 64 MORE
•1470  STA PLOT
•1480  BCC NEXT
•1490  INC PLOT+1
•1500  CLC
•1510 NEXT DEX
•1520  BNE BEGIN
•1530 PATCH LDX CHAR    ; PLOT=BASE
 +ROW*320
•1540  BEQ PATCH2
•1550 BEGIN2 LDA PLOT
•1560  ADC #8      ; ADD 8
•1570  STA PLOT
•1580  BCC NEXT2
•1590  INC PLOT+1
•1600  CLC
•1610 NEXT2 DEX
•1620  BNE BEGIN2
•1630 PATCH2 LDA PLOT    ; PLOT=PLO
 T+CHAR*8
•1640  ADC LINE     ; ADD 0-7
•1650  STA PLOT
•1660  BCC NEXT3
•1670  INC PLOT+1
•1680  CLC
•1690 NEXT3 LDX BITT ; PLOT=BYTE P
 LOT
•1700  LDY #0
•1710  LDA TABLE,X    ; LOOK UP BIT
•1720  ORA (PLOT),Y
•1730  STA (PLOT),Y   ; PLOT POINT
 1
•1740  DEX
•1750  LDA TABLE,X    ; LOOK UP BIT
 2
•1760  ORA (PLOT),Y
•1770  STA (PLOT),Y   ; PLOT POINT
 2
•1780 ; HERE COMES THE COLOR PLOT
•1790  LDX ROW
•1800  CLC
•1810  BEQ PATCH3
•1820 CONT LDA IND3
•1830.  ADC #40        ; ADD 40 (1 R
OW)
•1840  STA IND3        ; FOR EACH VA
LUE
•1850  BCC NEXT4       ; OF X
•1860  INC IND3+1
•1870  CLC
•1880 NEXT4 DEX
•1890  BNE CONT
•1900 PATCH3 LDA IND3
•1910  ADC CHAR        ; ADD 0-39
•1920  STA IND3
•1930  BCC CONT2
•1940  INC IND3+1
•1950 CONT2 LDA COLOR3
•1960  STA (IND3),Y
•1970  RTS             ; COLOR IS PLO
 TTED
•1980 ;
•1990 ; PATCHES ARE FOR SPECIAL CA
SES
•2000 ; WHICH I HADN'T EXPECTED.
•2010 ; THEY SIMPLY ALLOW A ZERO (
0)
•2020 ; VALUE TO BE SKIPPED OVER
•2030 ; WHICH ALLOWS FOR ACCURACY
•2040 ; IN THE TOP 8 PIXELS AND IN
 THE
•2050 ; COLUMN AT THE EXTREME LEFT
.
```

CALL PLOT

```
•1000 ; SET-UP ROUTINE FOR PLOT CA
LL
•1010 ; IT UNDERSTANDS WHAT THE CU
RRENT
•1020 ; COLOR IS AND IT BRANCHES
•1030 ; ACCORDINGLY.  THE SET-UP
•1040 ; PARAMETERS ARE X,Y COORDIN
ATE
•1050 ; AND THE CURRENT COLOR BEIN
G
•1060 ; USED (1-3) OR 0 FOR ERASIN
G.
•1070 ; THIS ROUTINE SETS UP THE O
R
•1080 ; TABLE ALONG WITH THE CORRE
```

```
CT
•1090 ; INDIRECT ADDRESSES.
•1100 ;
•1110 ; IT ALSO SETS UP THE AND TA
BLE.
•1120 ;
•1130 ; MAY 22,1983 BY PL
•1140 ;
•1150 ; CALL SET-UP WHEN PLOTTING
 POINTS
•1160 ;
•1170 *=$0360; STARTS IN MID CAS B
UFFER
•1180 CONUM  = $0350
•1190 TABLE  = $0342
•1200 TABLE2 = $0354
•1210 R3     = $0800
•1220 R2     = $0900
•1230 R1     = $0A00
•1240 R0     = $0B00
•1250 PLOT   = $F9
•1260 IND3   = $FB
•1270 IND    = $FD
•1280 XCO    = $033C
•1290        ASL XCO          ; DOU
BLE POINT FOR PLOT
•1300        LDA TABLE
•1310        CMP #1
•1320        BEQ CONT
•1330        LDX #0
•1340 ORCO   LDA OR,X
•1350        STA TABLE,X
•1360        LDA AEND,X       ; AEN
D = AND
•1370        STA TABLE2,X
•1380        INX
•1390        CPX #8
•1400        BNE ORCO
•1410 CONT   LDY #0
•1420        STY PLOT
•1430        LDA #32
•1440        STA PLOT+1
•1450        LDA CONUM
•1460        BEQ Z            ; Z =
 ZERO (0)
•1470        CMP #3
•1480        BEQ T            ; T =
 THREE (3)
•1490        STY IND
•1500        LDX #4
•1510        STX IND+1
•1520        CMP #1
•1530        BEQ O            ; O =
 ONE (1)
•1540        JMP R2           ; HAS
 TO BE TWO (R2)
•1550 O      JMP R1
•1560 Z      JMP R0
•1570 T      STY IND3
•1580        LDA #216
•1590        STA IND3+1
•1600        JMP R3
•1610 ;      TABLES FOLLOWS
•1620 OR     .BYTE 1,2,4,8,16,32,6
4,128
•1630 AEND   .BYTE 254,253,251,247
,239,223,191,127
```

CIR SCREEN

```
•1000 ;  THIS IS A SIMPLE FILL ROU
TINE
•1010 ;  TO CLEAR OUT THE HI-RES S
CREEN
•1020 ;
•1030 ;  MAY 29,1983 BY PL
•1040 ;
•1050 *=$0C00
•1060 SCREEN = $2000
•1070 COL3   = $D800
•1080 COL12  = $0400
•1090 BACK   = $D021
•1100 TEMP   = $02B0
•1110        LDY #0
•1120        LDX #32
•1130        STX START+2      ; THI
S RESETS THE
•1140        INX              ; ERA
SING POINTERS
•1150        STX START+5
•1160        INX
•1170        STX START+8
•1180        INX
•1190        STX START+11
•1200        LDA BACK         ; THI
S THROWS THE
•1210        ASL A            ; BAC
KGROUND COLOR
•1220        ASL A            ; INT
O ALL COLOR
•1230        ASL A            ; REG
ISTERS (I HAVE
•1240        ASL A            ; THE
 SAME COLOR IN
•1250        STA TEMP         ; THE
 UPPER AND LOWER
•1260        LDA BACK         ; 4 B
```

```
 ITS ON SCREEN)
•1270          AND #15
•1280          ORA TEMP
•1290 LOOP     STA COL3,Y          ; HER
 E'S WHERE THE
•1300          STA COL3+256,Y      ; COL
 OR INFO GETS
•1310          STA COL3+512,Y      ; PUT
  INTO COLOR
•1320          STA COL3+744,Y      ; AND
  SCREEN RAM
•1330          STA COL12,Y
•1340          STA COL12+256,Y
•1350          STA COL12+512,Y
•1350          STA COL12+512,Y
•1360          STA COL12+744,Y
•1370          INY
•1380          BNE LOOP
•1390          TYA
•1400 START    STA SCREEN,Y        ; HER
 E'S WHERE
•1410          STA SCREEN+256,Y     ; T
 HE HI-RES,
•1420          STA SCREEN+512,Y     ; M
 ULTI-COLORED
•1430          STA SCREEN+768,Y     ; P
 AGE GETS
•1440          INY                  ; A
 LL CLEARED
•1450          BNE START
•1460          LDX START+2
•1470          CPX #60             ; TES
 T FOR END OF CLEAR
•1480          BEQ END             ; IF
 YES, THEN END
•1490          INC START+2         ; IF
 NOT, MODIFY CODE
•1500          INC START+5         ; ON
 THE RUN
•1510          INC START+8
•1520          INC START+11
•1530          JMP START           ; BAC
 K TO LOOP
•1540 END      RTS
```

COLOR FILL

```
•1000 ; THIS IS THE FILL COLOR ROU
 TINE
•1010 ; IT SIMPLY CHANGES EVERY PO
 INT
•1020 ; ON THE SCREEN WITH A CERTA
 IN BIT
•1030 ; TO A CERTAIN COLOR
•1040 ; COLOR # FOR CHANGING IS AT
  $2A2
•1050 ; COLOR TO CHANGE TO IS AT $
 2A3
•1060 ; THIS ROUTINE ONLY WORKS IN
  COLOR
•1070 ; RAM, IT DOES NOTHING TO TH
 E BIT
•1080 ; MAP....
•1090 ;
•1100 ; MAY 30,1983 BY PL
•1110 ;
•1120 *=$0D00
•1130 COL12  = $0400
•1140 COL3   = $D800
•1150 COLNUM = $02A2
•1160 FILCOL = $02A3
•1170 TEMP   = $02A4
•1180          LDY #0
•1190          LDA FILCOL
•1200          LDX COLNUM          ; LOA
 D X WITH COLOR #
•1210          CPX #1              ; SEE
  IF 1,2,OR 3
•1220          BEQ ONE             ; BRA
 NCH ACCORDINGLY
•1230          CPX #2
•1240          BEQ TWO
•1250 THREE    STA COL3,Y          ; ROU
 TINE 3 IS A
•1260          STA COL3+256,Y      ; SIM
 PLE FILL LOOP
•1270          STA COL3+512,Y
•1280          STA COL3+768,Y
•1290          INY
•1300          BNE THREE
•1310          RTS
•1320 ONE      ASL A               ; FOR
  1 YOU MUST SHIFT
•1330          ASL A               ; THE
  COLOR 4 BITS TO THE
•1340          ASL A               ; LEF
 T SINCE ONLY THE
•1350          ASL A               ; UPP
 ER 4 BITS ARE USED
•1360          STA TEMP            ; IN
 COLOR 1
•1370 ;
•1380 LOOP1    LDA COL12,Y         ; THI
 S LOOP
•1390          AND #15             ; SIM
 PLY STORES THE
•1400          ORA TEMP            ; CUR
```

```
 RENT COLOR IN
•1410         STA COL12,Y      ; THE
 UPPER 4 BITS
•1420         LDA COL12+256,Y ; OF
 SCREEN RAM
•1430         AND #15          ; WHI
 LE LEAVING
•1440         ORA TEMP         ; THE
 LOWER 4 BITS
•1450         STA COL12+256,Y ; THE
 SAME.
•1460         LDA COL12+512,Y
•1470         AND #15
•1480         ORA TEMP
•1490         STA COL12+512,Y
•1500         LDA COL12+744,Y
•1510         AND #15
•1520         ORA TEMP
•1530         STA COL12+744,Y
•1540         INY
•1550         BNE LOOP1
•1560         RTS
•1570 ;
•1580 TWO     STA TEMP
•1590 LOOP2   LDA COL12,Y      ; THI
 S ROUTINE
•1600         AND #240         ; ALM
 OST THE SAME
•1610         ORA TEMP         ; AS
 #1 EXCEPT THAT
•1620         STA COL12,Y      ; NOW
 THE LOWER
•1630         LDA COL12+256,Y ; 4 B
 ITS GET
•1640         AND #240         ; CHA
 NGED AND THE
•1650         ORA TEMP         ; UPP
 ER 4 STAY THE
•1660         STA COL12+256,Y ; SAM
 E.
•1670         LDA COL12+512,Y
•1680         AND #240
•1690         ORA TEMP
•1700         STA COL12+512,Y
•1710         LDA COL12+744,Y
•1720         AND #240
•1730         ORA TEMP
•1740         STA COL12+744,Y
•1750         INY
•1760         BNE LOOP2
```

JOY READ

```
•1000 ; THIS IS THE JOYSTICK READ
 PROGRAM
•1010 ; IT READS PORT 2 FOR MOVEME
 NT AND
•1020 ; RETURNS THE DIRECTION IN $
 02A7
•1030 ;
•1040 ; MAY 30,1983 BY PL
•1050 ;
•1060 PORTB   = $DC00
•1070 STORE   = $02A7
•1080 *=$0E00
•1090 ;
•1100         LDA #0
•1110         STA STORE
•1120 UP      LDA PORTB
•1130         ROR A
•1140         BCS DOWN         ; SEE
 IF JOY IS UP
•1150         LDX #2
•1160         STX STORE
•1170         ROR A
•1180         JMP LEFT         ; SIN
 CE UP SKIP DOWN
•1190 DOWN    ROR A
•1200         BCS LEFT         ; TES
 T DOWN
•1210         LDX #4
•1220         STX STORE
•1230 LEFT    ROR A
•1240         BCS RIGHT        ; TES
 T LEFT
•1250         LDA STORE
•1260         ADC #6
•1270         STA STORE
•1280         RTS              ; SIN
 CE LEFT SKIP RIGHT
•1290 RIGHT   ROR A
•1300         BCS END          ; TES
 T RIGHT
•1310         LDA STORE
•1320         ADC #3
•1330         STA STORE
•1340 END     RTS              ; IN
 ANY CASE, RETURN
•1350 ;
•1360 ; AT RTS TIME THE NUMBER IN
 STORE
•1370 ; CORRESPONDS TO THE JOYSTIC
 K
•1380 ; ARRAY IN THE MULTI-DRAW PR
 OGRAM
```

RSI INTERRUPT

```
•1000 ; THIS IS THE SPLIT SCREEN I
NTERRUPT
•1010 ; ROUTINE WHICH ALLOWS MULTI
-COLOR
•1020 ; MODE ON THE TOP 22 LINES A
ND
•1030 ; NORMAL TEXT ON THE BOTTOM
3.
•1040 ; THE MAIN INTERRUPT IS LOCA
TED
•1050 ; AT $0F00 WITH ITS SETUP RO
UTINE
•1060 ; AT $0E80. NEXT ISSUE THIS
WILL
•1070 ; BE DEALT WITH IN MORE DETA
IL
•1080 ; BUT FOR NOW, CHANGING $D01
2
•1090 ; CONTROLS THE BOUNDARIES OF
•1100 ; THE NORMAL AND HI-RES SCRE
ENS.
•1110 ;
•1120 ; BY CHANGING THAT VALUE YOU
•1130 ; CAN ENLARGE THE TEXT AREA
OR
•1140 ; EVEN MAKE WINDOWS OF TEXT
OR
•1150 ; GRAPHICS.  KEEP EXPERIMENT
ING !
•1160 ;
•1170 ; MAY 30,1983 BY PL
•1180 ;
•1190 *=$0E80; SET UP INT ROUTINE
•1200 INTLO  = $0314
•1210 INTHI  = INTLO+1
•1220 HIRES  = $D011
•1230 RASTER = HIRES+1
•1240 MULTI  = HIRES+5
•1250 RASBIT = HIRES+8
•1260 BACK   = $D021
•1270 MULBA  = $02C0
•1280 COUNT  = $02B2
•1290 ;
•1300        SEI              ; STO
P ALL INTS WHILE
•1310        LDA #15          ; CHA
NGING IRQ VECTOR
•1320        STA INTHI
•1330        LDA #0
•1340        STA INTLO
•1350        STA COUNT        ; RES
ET ALL VIC CHIP
•1360        STA RASTER       ; RAS
TER VALUES
•1370        LDA HIRES        ; ( W
HICH WILL BE
•1380        AND #127         ;   E
XPLAINED MORE
•1390        STA HIRES        ;   N
EXT ISSUE )
•1400        LDA RASBIT
•1410        ORA #1
•1420        STA RASBIT
•1430        CLI              ; ALL
OW INT TO OCCUR
•1440        RTS
•1450 ;
•1460 *=$0F00; HERE IS THE INT ROU
TINE
•1470 ;
•1480 BEGINT = $EA31
•1490 ENDINT = $EA81
•1500 BASEAD = HIRES+7
•1510 TEXTBA = $02B1
•1520        LDA COUNT        ; SEE
 WHERE RASTER
•1530        BNE TEXT         ; IS
LOCATED
•1540 ;
•1550 MULCOL LDA #226         ; NOW
 ITS AT TOP OF
•1560        STA RASTER       ; SCR
EEN, AND IT'S
•1570        LDA #59          ; RES
ETTING FOR
•1580        STA HIRES        ; LAT
ER ON
•1590        LDA MULBA
•1600        STA BACK
•1610        LDA #29
•1620        STA BASEAD
•1630        LDA #24
•1640        STA MULTI
•1650        LDA #1
•1660        STA RASBIT
•1670        STA COUNT
•1680        JMP BEGINT       ; RES
ET-JUMP TO INT
•1690 ;
•1700 TEXT   LDA #252         ; NOW
 AT BOTTOM
•1710 ;
•1720        STA RASTER       ; RES
ET FOR JUMP
•1730        LDA #27          ; BAC
K TO TOP
```

```
•1740          STA HIRES
•1750          LDA TEXTBA        ; ALL
  THIS EXPLAINED
•1760          STA BACK          ; NEX
 T TIME
•1770          LDA #21
•1780          STA BASEAD
•1790          LDA #8
•1800          STA MULTI
•1810          LDA #1
•1820          STA RASBIT
•1830          LDA #0
•1840          STA COUNT
•1850          JMP ENDINT        ; JUM
 P TO END OF INT
•1860 ;
•1870 ; THE SECOND INTERRUPT JUMPS
  TO
•1880 ; THE END OF THE NORMAL IRQ
•1890 ; ROUTINE BECAUSE IT ALREADY
•1900 ; JUMPED THERE DURING THIS S.
 CREEN
•1910 ; FRAME. SINCE THE RASTER UP
 DATES
•1920 ; THE SCREEN ONCE EVERY 60TH
  OF A
•1930 ; SECOND ONE JUMP TO THE NOR
 MAL
•1940 ; IRQ PER SCREEN PASS GIVES
 THE
•1950 ; MACHINE THE ILLUSION OF WO
 RKING
•1960 ; UNDER A NORMAL INTERRUPT S
 TATE.
•1970 ; (BUT WE KNOW BETTER !!!!!!
 )
```

PHONE BOOK

From page 28

```
•10 PRINT"{SC}PHONE BOOK -- RELATI
 VE FILE DEMO":OPEN15,8,15,"I0"
•20 PRINT"{CD}{CD}SETUP NEW FILE (
 Y/N)?";
•30 GETA$:IFA$<>"N"ANDA$<>"Y"THEN3
 0
•40 IFA$="N"THENPRINT"NO":GOTO60
•50 PRINT"YES":OPEN3,8,3,"@0:PHONE
 FILE,L,"+CHR$(40):GOTO100
•60 OPEN7,8,7,"COUNTER,S,R":INPUT#
 7,N:CLOSE7:OPEN3,8,3,"PHONEFILE"
•100 PRINT"{SC}CHOOSE ONE:":PRINT"
 1- ENTER NAME, NUMBER":PRINT"2- S
 EARCH FOR NAME"
•110 PRINT"3- DISPLAY ALL ENTRIES"
 :PRINT"4- EXIT"
•120 PRINT"{CD}ENTER CHOICE-->";
•130 GETA$:A=VAL(A$):IFA<1ORA>4THE
 N130
•140 ONAGOTO200,400,600,800
•200 PRINT"{SC}ACTIVE ENTRIES ="N
•205 IFN=255THENPRINT"NO MORE ROOM
  IN FILE.":GOSUB990:GOTO100
•210 PRINT"{CD}ENTER NAME, NUMBER:
 ":I$="":J$=I$:INPUTI$,J$:IFI$=""O
 RJ$=""THEN100
•220 IFLEN(I$)+LEN(J$)>37THENPRINT
 "ENTRY IS TOO LONG...":GOTO210
•230 N=N+1:PRINT#15,"P"CHR$(3)CHR$
 (N)CHR$(0):PRINT#3,I$","J$:GOTO10
 0
•400 IFN=0THENPRINT"NO ENTRIES IN
 FILE.":GOSUB990:GOTO100
•405 PRINT"{SC}ENTER NAME TO SEARC
 H FOR:":N$="":INPUTN$:IFN$=""THEN
 100
•410 J=0:FORX=1TON:PRINT#15,"P"CHR
 $(3)CHR$(X)CHR$(0):INPUT#3,I$,J$
•415 IFI$<>N$THEN430
•420 X=N:J=N
•430 NEXT:IFJ=0THENPRINT"NOT FOUND
 .":GOSUB990:GOTO100
•440 PRINTN$,J$:GOSUB990:GOTO100
•600 PRINT"{SC}HIT F1 TO PAUSE AND
  CONINUE{CD}"
•610 IFN=0THENPRINT"NO ENTRIES IN
 FILE.":GOSUB990:GOTO100
•620 FORX=1TON:PRINT#15,"P"CHR$(3)
 CHR$(X)CHR$(0):INPUT#3,I$,J$:PRIN
 TI$,J$
•625 GETA$:IFA$<>"{F1}"THEN630
•627 GETA$:IFA$<>"{F1}"THEN627
•630 NEXT:GOSUB990:GOTO100
•800 PRINT"{SC}EXIT, ARE YOU SURE
 (Y/N)? ";
 810 GETA$:IFA$<>"N"ANDA$<>"Y"THEN
•810
•820 IFA$="N"THENPRINT"NO":GOTO100
•830 PRINT"YES":OPEN7,8,7,"@0:COUN
 TER,S,W":PRINT#7,N:CLOSE7:INPUT#1
 5,A,B$,C,D
•840 CLOSE15:END
•990 PRINT"{CD}HIT RETURN TO CONTI
 NUE."
```

```
995 GETA$:IFA$<>CHR$(13)THEN995
999 RETURN
```

MUSIC MAKER PART TWO

From page 70

VOICE 1 (Blocks 1-100)

```
BLOCK 1 DURATION 31  4 G
BLOCK 2 DURATION 1  REST
BLOCK 3 DURATION 7  4 G
BLOCK 4 DURATION 1  REST
BLOCK 5 DURATION 7  4 E
BLOCK 6 DURATION 1  REST
BLOCK 7 DURATION 7  4 G
BLOCK 8 DURATION 1  REST
BLOCK 9 DURATION 7  5 C
BLOCK 10 DURATION 1  REST
BLOCK 11 DURATION 31  4 G
BLOCK 12 DURATION 1  REST
BLOCK 13 DURATION 7  4 G
BLOCK 14 DURATION 1  REST
BLOCK 15 DURATION 7  4 E
BLOCK 16 DURATION 1  REST
BLOCK 17 DURATION 7  4 G
BLOCK 18 DURATION 1  REST
BLOCK 19 DURATION 7  5 C
BLOCK 20 DURATION 1  REST
BLOCK 21 DURATION 15  4 G
BLOCK 22 DURATION 1  REST
BLOCK 23 DURATION 7  4 G
BLOCK 24 DURATION 1  REST
BLOCK 25 DURATION 7  5 C
BLOCK 26 DURATION 1  REST
BLOCK 27 DURATION 15  4 G
BLOCK 28 DURATION 1  REST
BLOCK 29 DURATION 7  4 G
BLOCK 30 DURATION 1  REST
BLOCK 31 DURATION 7  5 C
BLOCK 32 DURATION 1  REST
BLOCK 33 DURATION 15  4 G
BLOCK 34 DURATION 1  REST
BLOCK 35 DURATION 15  4 G# (AB)
BLOCK 36 DURATION 1  REST
BLOCK 37 DURATION 15  4 A
BLOCK 38 DURATION 1  REST
BLOCK 39 DURATION 15  4 B
BLOCK 40 DURATION 1  REST
BLOCK 41 DURATION 15  5 C
BLOCK 42 DURATION 1  REST
BLOCK 43 DURATION 15  REST
BLOCK 44 DURATION 1  REST
BLOCK 45 DURATION 31  REST
BLOCK 46 DURATION 1  REST
BLOCK 47 DURATION 64  REST
BLOCK 48 DURATION 64  REST
BLOCK 49 DURATION 40  REST
BLOCK 50 DURATION 15  4 G
BLOCK 51 DURATION 1  REST
BLOCK 52 DURATION 15  4 A
BLOCK 53 DURATION 1  REST
BLOCK 54 DURATION 15  4 B
BLOCK 55 DURATION 1  REST
BLOCK 56 DURATION 15  5 C
BLOCK 57 DURATION 1  REST
BLOCK 58 DURATION 32  REST
BLOCK 59 DURATION 15  4 B
BLOCK 60 DURATION 1  REST
BLOCK 61 DURATION 15  4 A
BLOCK 62 DURATION 1  REST
BLOCK 63 DURATION 32  REST
BLOCK 64 DURATION 15  4 E
BLOCK 65 DURATION 1  REST
BLOCK 66 DURATION 63  4 G
BLOCK 67 DURATION 1  REST
BLOCK 68 DURATION 16  REST
BLOCK 69 DURATION 15  4 G
BLOCK 70 DURATION 1  REST
BLOCK 71 DURATION 15  4 A
BLOCK 72 DURATION 1  REST
BLOCK 73 DURATION 15  4 B
BLOCK 74 DURATION 1  REST
BLOCK 75 DURATION 15  5 C
BLOCK 76 DURATION 1  REST
BLOCK 77 DURATION 32  REST
BLOCK 78 DURATION 15  4 B
BLOCK 79 DURATION 1  REST
BLOCK 80 DURATION 15  4 A
BLOCK 81 DURATION 1  REST
BLOCK 82 DURATION 32  REST
BLOCK 83 DURATION 15  4 E
BLOCK 84 DURATION 1  REST
BLOCK 85 DURATION 63  4 F
BLOCK 86 DURATION 1  REST
BLOCK 87 DURATION 15  REST
BLOCK 88 DURATION 1  REST
BLOCK 89 DURATION 15  4 G
BLOCK 90 DURATION 1  REST
BLOCK 91 DURATION 15  4 G
BLOCK 92 DURATION 1  REST
BLOCK 93 DURATION 15  4 G# (AB)
BLOCK 94 DURATION 1  REST
BLOCK 95 DURATION 15  4 A
BLOCK 96 DURATION 1  REST
BLOCK 97 DURATION 32  REST
```

```
BLOCK 98 DURATION 15  4 A# (BB)
BLOCK 99 DURATION 1  REST
BLOCK 100 DURATION 15  4 B
```

VOICE 2

(Blocks 1-100)

```
BLOCK 1 DURATION 31  5 G
BLOCK 2 DURATION 1  REST
BLOCK 3 DURATION 39  REST
BLOCK 4 DURATION 1  REST
BLOCK 5 DURATION 27  5 G
BLOCK 6 DURATION 3  REST
BLOCK 7 DURATION 35  REST
BLOCK 8 DURATION 1  REST
BLOCK 9 DURATION 15  5 G
BLOCK 10 DURATION 1  REST
BLOCK 11 DURATION 19  REST
BLOCK 12 DURATION 1  REST
BLOCK 13 DURATION 15  5 G
BLOCK 14 DURATION 1  REST
BLOCK 15 DURATION 15  REST
BLOCK 16 DURATION 1  REST
BLOCK 17 DURATION 15  5 G
BLOCK 18 DURATION 1  REST
BLOCK 19 DURATION 15  5 F
BLOCK 20 DURATION 1  REST
BLOCK 21 DURATION 15  5 E
BLOCK 22 DURATION 1  REST
BLOCK 23 DURATION 15  5 F
BLOCK 24 DURATION 1  REST
BLOCK 25 DURATION 15  6 C
BLOCK 26 DURATION 1  REST
BLOCK 27 DURATION 7  5 G
BLOCK 28 DURATION 1  REST
BLOCK 29 DURATION 7  5 G
BLOCK 30 DURATION 1  REST
BLOCK 31 DURATION 15  5 G
BLOCK 32 DURATION 1  REST
BLOCK 33 DURATION 15  5 G
BLOCK 34 DURATION 1  REST
BLOCK 35 DURATION 15  REST
BLOCK 36 DURATION 1  REST
BLOCK 37 DURATION 7  5 G
BLOCK 38 DURATION 1  REST
BLOCK 39 DURATION 7  5 G
BLOCK 40 DURATION 1  REST
BLOCK 41 DURATION 15  5 G
BLOCK 42 DURATION 1  REST
BLOCK 43 DURATION 15  5 G
BLOCK 44 DURATION 1  REST
BLOCK 45 DURATION 15  REST
BLOCK 46 DURATION 1  REST
BLOCK 47 DURATION 7  5 G
BLOCK 48 DURATION 1  REST
BLOCK 49 DURATION 7  5 G
BLOCK 50 DURATION 1  REST
BLOCK 51 DURATION 15  5 G
BLOCK 52 DURATION 1  REST
BLOCK 53 DURATION 15  5 G
BLOCK 54 DURATION 1  REST
BLOCK 55 DURATION 15  5 G
BLOCK 56 DURATION 1  REST
BLOCK 57 DURATION 15  REST
BLOCK 58 DURATION 1  REST
BLOCK 59 DURATION 31  REST
BLOCK 60 DURATION 1  REST
BLOCK 61 DURATION 20  REST
BLOCK 62 DURATION 1  REST
BLOCK 63 DURATION 15  6 C
BLOCK 64 DURATION 1  REST
BLOCK 65 DURATION 25  5 G
BLOCK 66 DURATION 5  REST
BLOCK 67 DURATION 16  REST
BLOCK 68 DURATION 1  REST
BLOCK 69 DURATION 15  5 G
BLOCK 70 DURATION 1  REST
BLOCK 71 DURATION 25  5 E
BLOCK 72 DURATION 5  REST
BLOCK 73 DURATION 14  REST
BLOCK 74 DURATION 1  REST
BLOCK 75 DURATION 15  5 E
BLOCK 76 DURATION 1  REST
BLOCK 77 DURATION 15  5 G
BLOCK 78 DURATION 1  REST
BLOCK 79 DURATION 13  6 C
BLOCK 80 DURATION 1  REST
BLOCK 81 DURATION 52  5 G
BLOCK 82 DURATION 2  REST
BLOCK 83 DURATION 26  REST
BLOCK 84 DURATION 1  REST
BLOCK 85 DURATION 15  6 C
BLOCK 86 DURATION 1  REST
BLOCK 87 DURATION 30  5 G
BLOCK 88 DURATION 2  REST
BLOCK 89 DURATION 15  REST
BLOCK 90 DURATION 1  REST
BLOCK 91 DURATION 15  5 G
BLOCK 92 DURATION 1  REST
BLOCK 93 DURATION 30  5 E
BLOCK 94 DURATION 2  REST
BLOCK 95 DURATION 15  REST
BLOCK 96 DURATION 1  REST
BLOCK 97 DURATION 15  5 F
BLOCK 98 DURATION 1  REST
BLOCK 99 DURATION 15  5 G
BLOCK 100 DURATION 1  REST
```

TABLE OF SIMONS' BASIC COMMANDS

From page 58

Programming Aids:

AUTO	Automatically generates line numbers at a specific increment
RENUMBER	Automatically renumbers all program lines (does not renumber GOSUB and GOTO)
KEY	Assigns a command to a function key
DISPLAY	Lists the commands assigned to the function keys
CGOTO	Computes the line number for a branch instruction
RESET	Moves data Pointers to a specific line of data
MERGE	Loads a saved program and appends it to the one in memory
PAGE	Divides a program listing into "Pages" of n lines
DELAY	Controls the rate of a program listing
OPTION	Highlights all of Simons' BASIC commands during a program listing
FIND	Searches a BASIC program and displays line numbers of each occurrence of the specified code or character string
PAUSE	Stops a program for a specified number of seconds and displays an optional message
TRACE	Displays the number of the program line being executed
RETRACE	Resumes TRACE after editing a program
DUMP	Displays values of all non-array variables
OLD	Reverses the NEW command
COLD	Resets the Commodore 64 to start of Simons' BASIC

Program Security:

DISAPA	Indicates a program line to be hidden
SECURE	Hides all program lines beginning with the DISAPA command

Text Manipulation:

INSERT	Inserts one character string into another
INST	Overwrites a string at a specified location
PLACE	Determines the position of a substring within a main string
DUP	Duplicates a character string a specified within a main string
CENTRE	Centers a character string on a screen line (yes, that is how it is spelled)
AT	Prints a character string at a specified screen coordinate
USE	Formats numeric data (similar to PRINT USING)

Input Validation Commands:

FETCH	Limits the type and number of characters allowed for input
INKEY	Tests for a function key input
ON KEY	Branches to a specific line number if any of the specified characters are pressed
DISABLE	Cancels the ON KEY command
RESUME	Reinstates the previous ON KEY command

Arithmetic Functions:

MOD	Returns the remainder of an integer division
DIV	Returns the integer value of a floating point division
FRAC	Returns up to nine decimal places of the fractional part of a floating point number
%	Converts from binary to decimal
$	Converts from hexadecimal to decimal
EXOR	Performs an exclusive or between two numbers

Disk Commands:

DISK	Opens the disk command channel and transmits a disk command
DIR	Performs a directory listing without affecting the program in memory

Graphics:

COLOUR	Sets screen background and border colors (yes, that is how it is spelled)
HIRES	Initializes high-resolution graphics mode and selects a plotting and background color
MULTI	Initializes multicolor graphics mode and selects three plotting colors
LOW COL	Changes colors from those set by the HIRES or MULTI commands
HI COL	Reverts to colors originally selected by the HIRES or MULTI command
REC	Draws a rectangle with specified dimensions at a specified coordinate
PLOT	Plots a dot at a specified coordinate
LINE	Draws a line between specified coordinates
CIRCLE	Draws a circular or elliptical shape at a specified coordinate
ARC	Draws an arc of a circular shape
ANGL	Draws the radius of a circle
PAINT	Fills an enclosed area with color
BLOCK	Draws a filled-in rectangle
DRAWS	Designs a shape
ROT	Displays a shape, created by draw, in a specified orientation and size
CSET	Selects either of the C-64 character sets or displays the last graphics screen
CHAR	Prints a character on a graphics screen
TEXT	Prints a character string on a graphics screen
TEST	Determines if something has been drawn at screen location
NRM	Returns to a low-resolution screen from a graphics screen

Screen Manipulation:

BCKGNDS	Specifies the background color of a shifted character, a reverse-field unshifted character or a reverse-field shifted character. Only characters on top of each key can be used. Graphics characters are not affected
FLASH	Flashes a specific screen color at a specified rate
OFF	Turns off the FLASH command
BFLASH	Flashes the screen border between specified colors at a specified rate
FCHR	Fills a specific area of the screen with a specified character
FILL	Fills a specified area of the screen with a specified character of a specified color
FCOL	Changes the color of all characters in a specified area of the screen
MOVE	Duplicates a section of the screen
INV	Inverts all characters in a specified area of the screen
LEFT, RIGHT, UP, DOWN-B or W	Scrolls a specified area of the screen in the specified direction with wrap around (W) or without wrap around (B). For example: LEFTB
SCRSV	Stores a low resolution screen on disk or tape
SCRLD	Loads a stored screen from disk or tape
COPY	Produces a copy of a graphics screen to the serial printer
HRDCPY	Prints a copy of a low resolution screen to the serial printer

Sprite and User Defined Graphics

DESIGN	Allocates memory for an MOV (sprite or moveable object block). Also used to specify custom characters with MEM command below

@	Sets up a design grid for an MOV (This command did not seem to be needed when designing the MOB. The example in the manual did not use it either). Also sets up the design grid for custom characters with the MEM command below
CMOB	Defines colors for a multi-color MOB
MOB SET	Initializes an MOB
MMOB	Displays and moves an MOB
RLOCHMOB	Moves a previously displayed MOB
DETECT	Initializes MOB collision detection
CHECK	Tests for an MOB collision
MOB OFF	Removes an MOB from the screen
MEM	Transfers character set from ROM to RAM

Structured Programming:

IF...THEN...ELSE	Tests for a condition and branches to one instruction if true and another if false
REPEAT...UNTIL	Performs a program loop until a specified condition is met
RCOMP	Re-executes the last IF...THEN...ELSE condition test
LOOP...EXIT IF...END LOOP	Performs a continous loop until a specified condition is met

Procedures — Calling of Subroutines by Name

PROC	Labels a program routine
END PROC	Indicates the end of a procedure
CALL	Transfers program execution to a named procedure (equivalent to GOTO)
EXEC	Jump to a named procedure and return when completed (equivalent to GOSUB)
LOCAL	Assignes variables to a specific program routine
GLOBAL	Restores original values to local variables

Error Trapping:

ON ERROR	Causes program to jump to a specified line number when a BASIC program error is found
OUT	Turns off the last ON ERROR command
NO ERROR	Re-enables BASIC's error handling routines

Music Commands:

VOL	Sets sound volume
WAVE	Selects voice waveform or sound type
ENVELOPE	Defines the shape of a sound, that is, the attack, decay, sustain and release parameters
MUSIC	Allows writing of music as an alphanumeric string
PLAY	Plays defined music while halting or continuing with program execution

Gameport Functions:

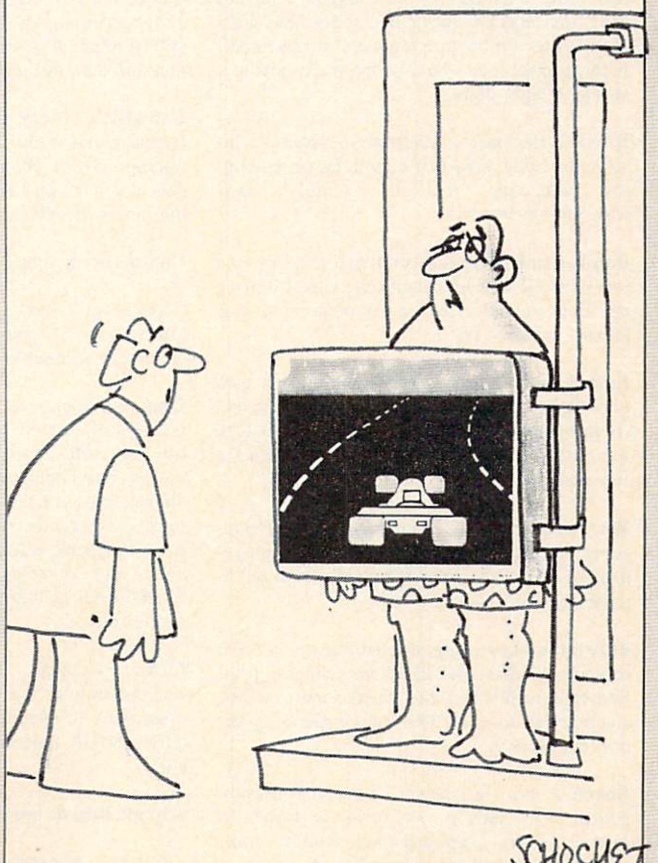
PENX	Returns the X coordinate of the light pen
PENY	Returns the Y coordinate of the light pen
POT	Returns the position of a paddle
JOY	Returns the joystick Position

Error Messages:

Simons' BASIC adds ten error messages to identify specific program errors.

?BAD MODE
?NOT HEX CHARACTER
?NOT BINARY CHARACTER
?UNTIL WITHOUT REPEAT
?END LOOP WITHOUT LOOP
?END PROC WITHOUT EXEC
?PROC NOT FOUND
?NOT ENOUGH LINES
?BAD CHAR FOR A MOB IN LINE n
?STACK TO LARGE

ADVERTISERS INDEX



"How's it look, Doc?"

Access. Retrieval of information from computer memory.

Access time. Time needed to retrieve information from computer memory.

Address. Location (identified by a number) of a byte of information in the computer's memory.

Alphanumeric. Alphabetic and numeric characters.

Analog signal. A continuous electronic signal of any frequency or strength that represents a condition (i.e.: the position of the game control paddles).

Applications software. Programs that will instruct the computer to perform either a specific task or a series of related tasks, usually relating to business or home uses.

ASCII. The American Standard Code for Information Interchange. This code allows two computers, which may entertain different languages, to communicate.

Assembler. A program that converts information in the computer's memory into a binary code for proper execution.

Assembly language. Machine language that has been translated into mnemonic codes, thus making it easier for the programmers to remember. A three letter code would be the equivalent to a string of eight digits.

BASIC. Beginner's All-purpose Symbolic Instruction Code. Easy to use, popular programming language that is widely used with microcomputers.

Baud. A unit (bits per second) that measures the rate of speed at which data is translated from one device to another. (Heck, every baud *we've* ever known has been fast.)

Binary code. A system of numbers that uses two digits, 0 and 1, to express all characters (both numbers and letters). The computer then uses this string of numbers to process information.

Bit. An acronym for *B*inary Dig*IT*. Represents either "0" or "1" in the binary code. Approximately eight bits (one byte) are required to represent one character.

Bidirectional printing. Special feature on some computer printers that allows the printer to print first right to left and then drop to the next line and print left to right. This feature speeds up the printing process.

Board. A thin, rectangular, flat electronic component that contains one or more layers of printed circuits. Chips and other electronic parts are often attached to a circuit board.

Boot. To start or restart a computer by transferring instructions ("Heel!"?) from a storage device, such as a tape or disk, into the computer's memory.

Bug. A logic error in a computer program that causes the program to dysfunction.

Bus. A conductor that allows data to be passed between the various computer components. Buses are manufactured that will allow different brands of components to be used with the same computer. (They can often be seen entering terminals.)

Byte. Plural of bit. There are eight bits in one byte. Each byte represents one character.

CAI. Computer Assisted Instruction. This refers to a variety of instructional and/or educational software.

Card. A circuit board which is attached to a standing computer, often to boost its memory capability.

Cartridge. Sometimes referred to as a ROM module or solid state cartridge. Device that contains a prerecorded program (such as a game).

Cassette Tape Recorder. Device used either to store data or to house prerecorded programs. Often, but not always, the same type used in audio recording. If so, a special interface may still be required. A much slower and less reliable medium than disk drive.

C-BASIC. Very popular non-interactive language that is utilized by 8080, 8085 and Z80 microprocessor computers. C-BASIC executes data at a more accelerated rate than the standard interpreter BASIC.

Character. A single symbol, letter, or number.

Character set. Total catalogue of alphanumeric, special, and punctuation characters accessible to a printer or terminal.

Chip. An integrated circuit, a quarter to three quarters of an inch square, that is etched on a tiny piece of silicon. Chips are the building blocks of computers and are able to contain anywhere from a few dozen to several thousand transistors or circuit elements. They perform many functions: calculations, memory operation and storage or controlling other chips. Up to 32,000 bytes of information may be stored in a single chip.

CPU. Central Processing Unit, or microprocessor. The electronic "core" of a computer. All information is passed through the CPU and all functions are executed from the CPU.

Circuit Board. see board.

COBOL. Acronym for Common Business Oriented Language. High level language for business applications.

Command. An instruction (e.g.: run, load) that tells the computer to do something.

Compatibility. The ability of the computer and any or all of its peripherals to be able to function in conjunction with one another.

Compiler. A program that translates any high level language into the binary code that is required for the computer.

Computer. A programmable electronic device, consisting of a CPU, memory and input/output (I/O) capability, that stores, retrieves, and processes data.

Computer system. A computer setup that consists of a computer terminal and its software and various peripherals.

CP/M. Control program for microcomputers. A single-user operating system for microcomputers that is in very wide use, especially for business applications.

CRT. Cathode Ray Tube. The most popular type of display screen. Often it is simply the screen of an ordinary television set though it can be a monitor specially designed for use with computers.

Cursor. Symbol, most often a small flashing square, that indicates where the next character will appear.

Data. All information that is processed by the computer.

Datum. Singular of data.

Data Bank. The central storage locations for all of the information that the computer has access to.

Data base manager. Program that enables user to enter files, organize matter and sort and retrieve information.

Daisy wheel printer. A letter quality impact printer that utilizes a daisy wheel, a circular printing element containing as many as a hundred characters.

Descenders. The lower case characters g,j,p,q, and y, which have tails that extend past the baseline formed by the other characters.

Disk. A flat, round device for storing computer data. Like a phonograph disc, it rotates and contains information in tracks. Unlike a phonograph disc, it is never removed from its protective jacket. A disk can be made of hard plastic (hard disk) or soft plastic (floppy disk, diskette).

Diskette. See disk.

Disk drive. A machine that rotates a disk for the purpose of entering new information or copying existent information from the disk into the com-

puter's memory.

Display. Generally a television-style screen, it shows the user the information contained in the terminal memory.

Documentation. Written instructions for operating hardware or software.

DOS. Disk Operating System. (See Operating System.)

Dot matrix. The method by which most printers form alphanumeric characters or graphics, by printing a series of dots.

Double-density. Refers to the process of doubling the amount of information that can be placed on a disk or cassette.

Downtime. When a computer is "down," or unoperational.

Dual Intensity. Refers to a printer that can produce bold type in additional to ordinary type.

Editor. A program which makes it possible to enter text into a computer.

Electronic mail. Telephone transmission (via a modem) of information between computer users.

EPROM. Erasable Programmable Read Only Memory. A device such as a cartridge or disk that can have data added to it or erased.

Floppy disk. See disk.

FORTRAN. Formula Translator. Advanced programming language used for complex mathematical operations.

Emulator. A program-translating device that enables software designed for a particular computer to run on a different one.

Firmware. ROM-contained data, built into the machine or added via software, that cannot be changed.

Flow chart. Diagram outlining procedure for writing a program.

Format. (or Initialize) To get a disk ready to accept information.

Function key. A key on a computer that can be assigned to perform a particular function.

Full duplex. Communication mode capable of simultaneous transmission and reception of data.

Graphics. Non-textual CRT displays: charts, symbols, etc.

Graphics tablet. A device which will transfer whatever is drawn on it onto the computer monitor.

Half duplex. Communication mode capable of transmission of data and reception of data, but not at the same time.

Hard copy. Information, when it has been transferred from memory to a computer printout, is hard copy.

Hardware. Mechanical, non-software components of a computer system: computer, disk drive, printer, etc.; also, the components of the components, such as transistors, circuits, etc. Also the components of the components of the . . .just kidding.

Hard disk. Nonremovable storage device that is faster and has a far greater storage capacity than a floppy disk. It is generally hermetically sealed within the unit.

Hexadecimal. A base-16 numeration method widely used with computers. Numbers run from 0 to 9, then from A to F. Hexadecimal (or "hex") numbers are identified with the suffix H.

High-level language. An easily learned programming language that resembles human languages such as English. Examples. BASIC. COBOL, FORTRAN. Low level languages (such as Assembly) require that every command and function be separately programmed.

High resolution. High-quality graphics capability when applied to a video terminal. An individual software program may itself contain high-resolution graphics, meaning detailed or colorful graphics. A subjective term and therefore frequently abused (especially on New Years Eve).

Impact printer. A typewriter-style printer that presses characters through a ribbon.

Index hole. A hole punched in a floppy disk that is used by the disk drive to locate the beginning of sector 0 on the disk.

Instruction. A command contained in a program.

Interface. A connecting device which can be electronic or can be contained in software, for making a computer compatible with a peripheral device.

I/O. Input/Output. The system of pathways which channel information into a computer (usually through a keyboard, joystick, mouse, light pen) and out of the computer (usually onto a screen or printer).

K. See "Kilobyte"

Kilobyte. See "K". Seriously, a kilobyte (or 1K) is equal to 1,024 bytes (often rounded off to 1,000 bytes), and is the measure of the memory hardware or software can contain (8K, 64K, etc., etc.). Obviously, the higher the number the more capable the hardware/software.

Letter-quality. Refers to a printer that uses formed characters, such as those contained on a daisy wheel. Compare dot matrix printer.

Library. Collection of computer programs.

Light Pen. A device that allows a user to draw line figures or make menu selections directly on the screen. Proper hardware port and software required.

Load. Entry of information into the computer from an external storage, such as cassette player or disk drive.

LOGO. Programming language useful for graphics. Primarily employed in young children's education, each command is built logically on the previous command.

Loop. Programming statement used to order repetition of a task. A particularly useful one is called a "fruitful loop."

Machine language. Language used by the computer, comprised of binary numbers, into which the computer must translate programming languages.

Megabyte. Equal to 1,048,576 bytes (1048 kilobytes). Often abbreviated Mb.

Megahertz. Electrical frequency measure equalling one million cycles per second. Often abbreviated MHz.

Membrane. A flat computer keyboard that has touch-sensitive areas for each character rather than full-travel or stepped keys.

Memory. The word used to indicate any given computer's capacity to store information (10K memory etc.). There are different kinds of memory (ROM, RAM) and, within any computer, different memory locations (for sound, screen displays etc.).

Menu. A screen display of programs, on a particular disk or options in a program.

Microcomputer. An integrated small computer. It contains a microprocessor, memory, and interfaces for inputting and outputting information. Perfect example? Commodore 64.

Microprocessor. The brains of the computer, where mathematics and logical functions are performed. Also called the CPU.

Modem. Modulator/Demodulator. A device that changes information in analog form into digital form, and vice versa, for the purpose of transmitting computer information across a telephone line.

Monitor. Screen for displaying computer information.

Mouse. An input device, usually containing a selection button. The user slides the mouse on the

desktop beside the computer, and the cursor will duplicate the mouse's movement on the screen. Used to make menu selections and re-arrange information.

MP/M. Multiprogramming control Program for Microprocessors. A variety of the CP/M operating system that can be used by several users at a time.

Multi-user system. A system with a central pool of data or applications that can be accessed simultaneously be several users.

Number crunching. Refers to a computer's carrying out intricate or dense arithmetic or numerical functions.

Nybble. Half a byte, or four bits. (Or two shaves and two haircuts.)

Object code. A code in binary form, produced by an assembler or compiler program, A source code, by contrast, must be translated by an assembler or compiler before it can be executed by the CPU.

Octal. A base-8 numeration method often used with microcomputers. Numbers run from 0 to 7, and are identified with the suffix Q.

On-line. The state of being hooked up to an active computer, as in the case of a printer or disk drive that is connected and operational.

Operating system. The program(s) that supplies the computer system with its operating information, including start-up instructions at the time the system is turned on and supervisory instructions each time a new program is loaded.

Output. The path system that channels information out of a computer (usually to a screen or printer).

Overstriking. A method of producing boldface type by directing the printer to hit a character more than once.

Parallel. An input/output system that submits 8 bits of data at a time. An interface would have to be installed between a computer's serial port and a parallel printer, for example.

PASCAL. A more sophisticated programming language than BASIC, using less memory and producing faster programs. Named after Blaise Pascal.

PEEK. A programming command generally meaning: examine (specified location) and report the value that is represented there.

Peripheral. A hardware accessory to a computer, such as a printer or a modem.

Pixel. Picture element. A dot of light on a TV or computer screen, the smallest light fragment that the computer can address. Graphics with high resolution are generally composed of very small, and therefore numerous pixels.

Plotter. Machine for printing lines or graphs.

POKE. A programming command that is used to place a new value into a specified memory location.

Printer. Machine used to print computer information onto paper. See dot matrix, letter quality, daisy wheel.

Program. As a noun: a set of instructions given to a computer to enable it to perform a particular function. As a verb: to input such information into a computer.

Programming language. A language used in the composition of a computer program.

PROM. Programmable Read Only Memory. A permanent storage system for data that can be programmed both by the manufacturer and by the user.

Quad-density. Refers to a two-sided, double-density disk which is able to store four times the information of a one-sided, normal density disk.

RAM. Random Access Memory. Volatile storage system for data that can be changed—added to, subtracted from, rearranged—by the user. Information stored in RAM must be saved before the system is shut off or it will be lost.

Read/write. Signifies that information can be both read from and written into memory (RAM or permanent storage).

Resolution. A measure of the sharpness of a CRT (cathode ray tube) picture. Expressed in pixels, it can refer to either the number of scanning lines on the terminal or the number of addressable pixels on the screen.

Reverse video. The displaying of dark characters on a light background.

RF Modulator. Used to change computer video signals into radio frequency signals that can be picked up by a TV antenna and displayed.

RGB. Video signal composed of red, green, and blue that has much higher resolution and brighter colors than the standard composite color TV signals. A monitor with three separate electron guns, rather than the single gun used by the average color TV, is required.

ROM. Read Only Memory. Data that is built into a computer or software, and cannot be changed, such as the information that operates the computer immediately after start-up. In a computergame, for example, backgrounds which do not change or are not affected by gameplay are ROM. Spaceships and projectiles, which constantly are updated, are RAM.

Save. Recording information from the computer's memory onto tape or a disk.

Scrolling. Moving up and down (or from left to right) through the displayed information contained in the computer's memory, only a certain portion of which can be displayed on the terminal screen at one time.

Sector. A part of the track of a storage disk.

Serial. An input or output system that submits one bit of data at a time.

Serial port. A computer's input or output port through which data is transmitted in serial fashion. This is generally done, with home computers, through an RS232C serial interface port.

Single-sided. A disk with only one side that can store data.

Software. Programs to run on a computer on tape, disk, or cartridge. Everything from *Donkey Kong* to *Data Base*.

Source code. Program written in English-like words via an editor program which needs to be translated (with an assembler or compiler) into a language the computer understands.

Sprite. A high resolution programmable object. Any graphic figure is a sprite; sprites can be changed and animated through commands in BASIC

Syntax. Rules of grammatical usage governing programming language, as with English and other languages.

Terminal. The screen that displays computer information.

Text. The words and numbers displayed on the CRT screen.

Tractor feed. Mechanism that holds fan-fold paper in place and moves it through the printer, using sprockets that fit into the holes on both sides of the paper.

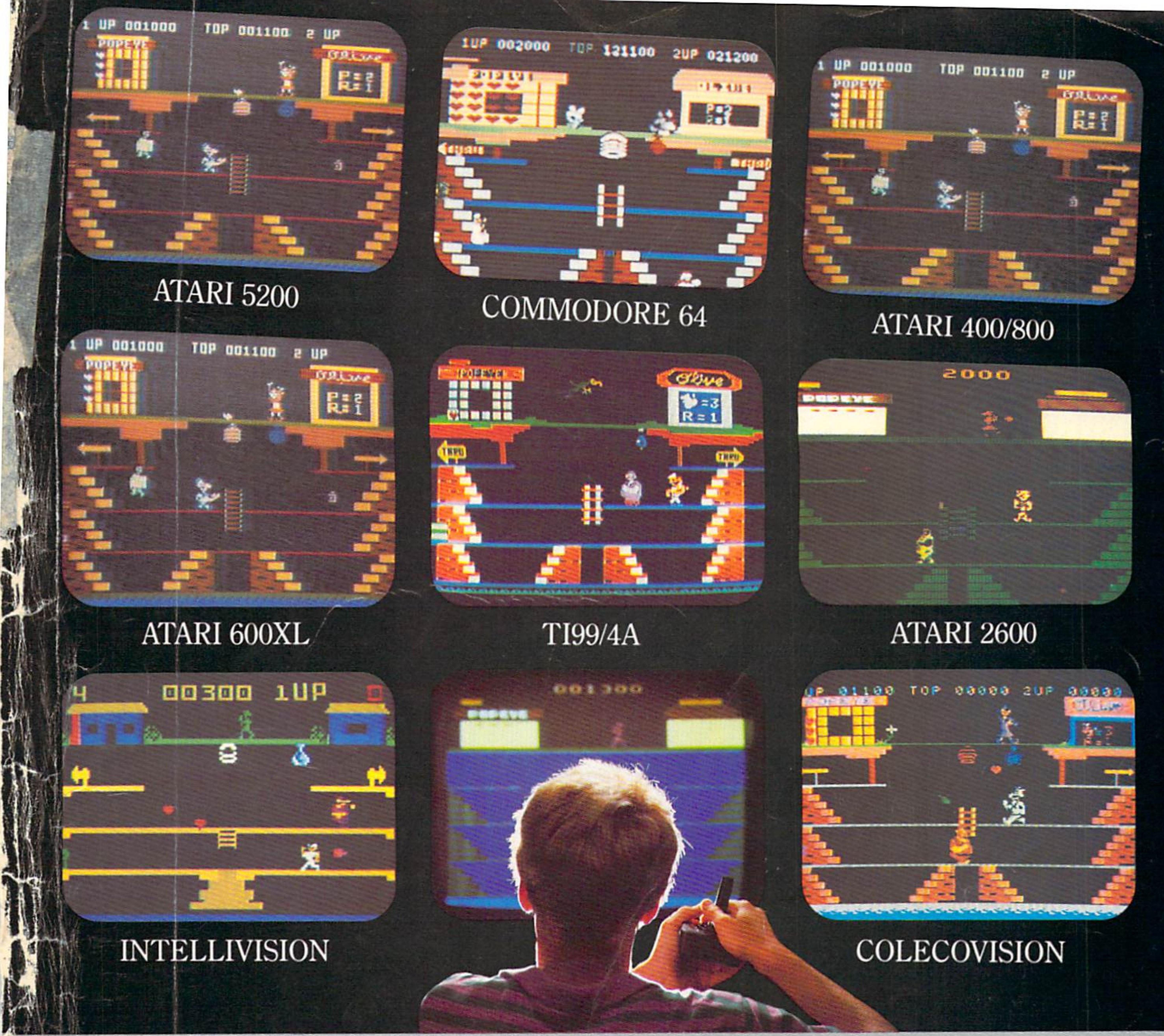
Volatile storage. A storage device, such as RAM, that loses the data it contains when power is cut off.

Winchester. A variety of hard disk that is sealed in a container.

Word processor. Can refer to the complete system, usually consisting of computer, display, memory storage, and printer, that is used to produce written documents; also refers to a software program that enables a computer to perform word processing functions.

Write. The process of transferring data from memory to permanent storage.

Write protection. Process that prevents writing to, or erasing from, a disk. With 8-inch disks, this is accomplished by removing a tab from the jacket; With 5¼ disks, by affixing the tab to the jacket.



EVERYONE HAS THEIR OWN SYSTEM FOR PLAYING POPEYE.®



Atari.® Intellivision.® ColecoVision.™ T.I.™ Commodore.™ Now you can play POPEYE, one of the most fun and challenging arcade games yet, on any one of them. Run through three screens of non-stop action, where you try to capture Olive Oyl's heart while avoiding untold dangers, including Brutus and the Sea Hag.

Run down to your local store for Popeye today. And while you're there, check out TUTANKHAM,™ FROGGER,™ Q*bert™ and SUPER COBRA,™ also from Parker Brothers' Arcade Action Series.

® PARKER BROTHERS



Reader Service No. 61

TURN YOUR COMMODORE-64 INTO THE HOTTEST ARCADE MACHINE IN TOWN



O'RILEY'S MINE™

You're a mad Irishman digging your way through an abandoned mine filled with oil, coal, gold, rubies, diamonds—and hungry creatures. You'll need the luck of the Irish to survive, but with so much at stake, it's a chance you're willing to take.



O'RILEY'S MINE™ available for Atari®, Apple® and Commodore 64®

MOON SHUTTLE™

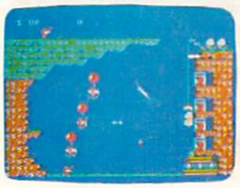
Your screen explodes with life-threatening action as you pilot your Moon Shuttle through outer space in this home version of the Nichibutsu® favorite. Awaiting you is The Prince of Darkness and his forces, which mysteriously multiply!



MOONSHUTTLE™ available for Atari®, Radio Shack Color®, Commodore 64®, coming soon for Apple®

POOYAN™

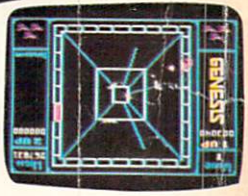
One of the biggest arcade game hits from Konami is a game of fast action in life-and-death encounters. You battle a pack of hungry wolves eager to catch your defenseless piglets. You'll need quick reflexes and a good arm in the new arcade hit from Datasoft.



POOYAN™ available for Atari®, Radio Shack Color®, Apple®, coming soon for Commodore 64®

GENESIS™

Balanced on the edge of disaster, you are the deadly King Scorpion defending your domain against fatally venomous spiders. Genesis captures all the intensity of Design Labs' arcade version of the first great battle on earth.



GENESIS™ available for Atari®, Apple®, IBM-PC®, and Commodore-64®.

Our reputation as one of the premier game manufacturers assures you of the highest quality attainable in action, strategy and graphics. For the hottest titles in entertainment, keep your eye on the leader—DATASOFT.

You'll Find The Best In Home Computer Software from Datasoft®

9421 Winnetka Avenue, Chatsworth, CA 91311 (213) 701-5161

ooyan™ is a trademark of Konami Industries Company, Ltd.
'Riley's Mine® and Datasoft® are registered trademarks f Datasoft Inc.

Moon Shuttle® is a registered trademark of Nichibutsu, USA.
GENESIS™ and Design Labs are trademarks of Design Labs.

Reader Service No. 54